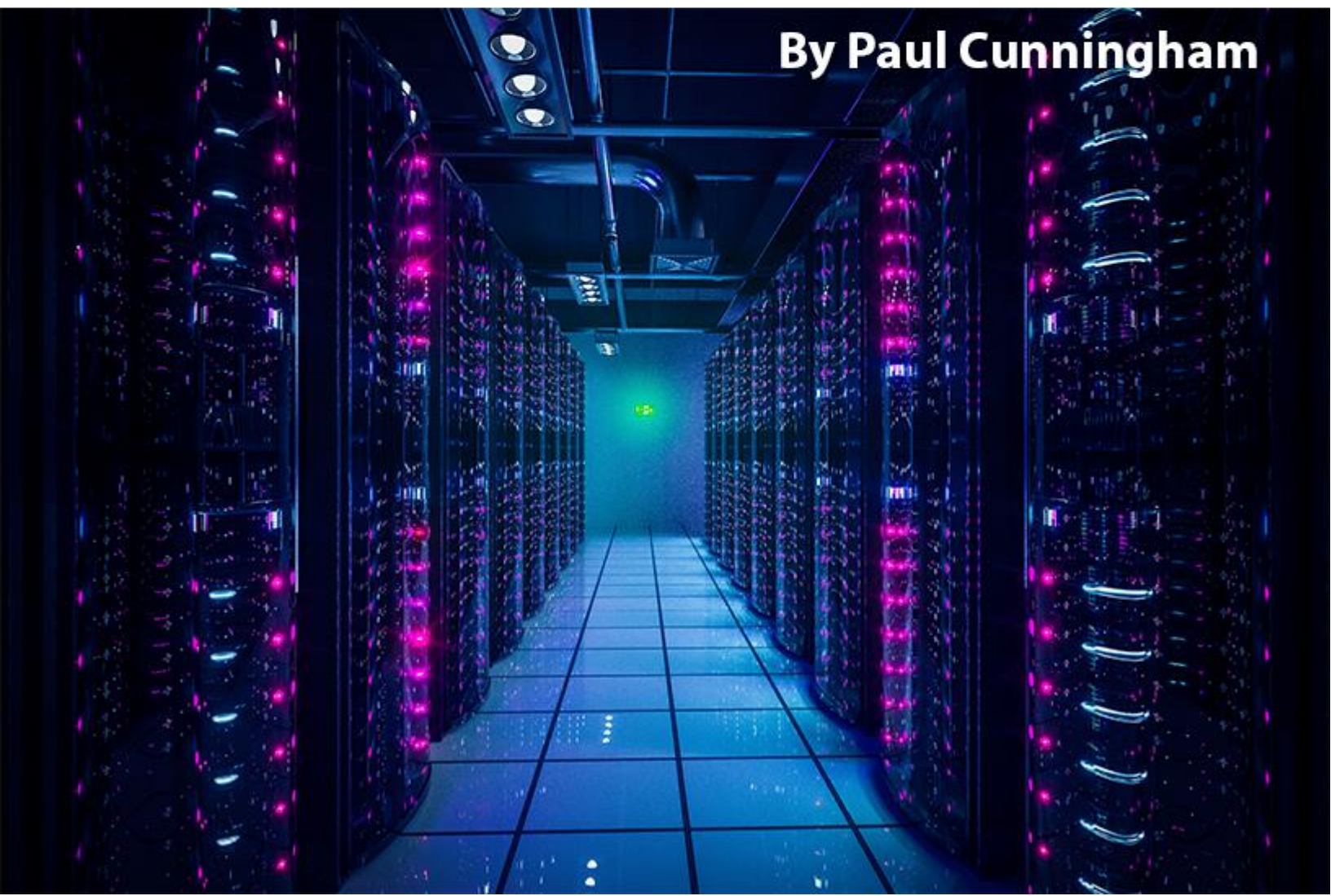


Exchange Server 2016 Quick Start Guide

By Paul Cunningham



© Copyright 2015 Paul Cunningham, LockLAN Systems Pty Ltd

The right of Paul Cunningham, LockLAN Systems Pty Ltd to be identified as author and copyright owner of this work is asserted by Paul Cunningham, LockLAN Systems Pty Ltd in accordance with Australian copyright laws as determined by the Australian Copyright Council.

Copyright extends to any and all countries in which this publication is purchased and/or viewed and/or read.

The reader of this publication indemnifies Paul Cunningham and LockLAN Systems Pty Ltd and its directors, officers, employees and agents from and against all losses, claims, damages and liabilities which arise out of any use of this publication and/or any application of its content.

About the Author



Paul Cunningham is a Microsoft MVP and Consultant living in Brisbane, Australia. He specializes in Exchange Server and Office 365, and is the founder of ExchangeServerPro.com, a leading website in the Exchange Server community.

Connect with Paul on [Twitter](https://twitter.com/ExchServerPro) (@ExchServerPro).

Table of Contents

| | |
|--|----|
| Welcome | 1 |
| Introduction to Exchange Server 2016 | 3 |
| Key Features and Concepts..... | 3 |
| Server Roles Architecture | 5 |
| What's New for Exchange Server 2013 Customers | 6 |
| What's New for Exchange Server 2010 Customers | 6 |
| Design Fundamentals..... | 7 |
| The Preferred Architecture | 7 |
| Sizing Guidance | 7 |
| Virtualization and Storage | 8 |
| System Requirements | 8 |
| Operating Systems | 8 |
| Active Directory | 9 |
| Exchange Organization | 9 |
| Network and Firewall..... | 10 |
| Installing Exchange Server 2016 | 11 |
| Which Edition of Exchange Server 2016 to Deploy? | 11 |
| Installing Exchange Server 2016 Pre-Requisites on Windows Server 2012 R2 | 14 |
| Running Exchange Server 2016 Setup | 15 |
| Managing Exchange Server 2016..... | 19 |
| Introduction to Management Tools..... | 19 |
| Role-Based Access Control..... | 21 |
| What State is the Exchange 2016 Server in Right Now? | 22 |
| Creating a User and Mailbox..... | 22 |

| | |
|--|----|
| Accessing the New User's Mailbox | 23 |
| Configuring Client Access..... | 27 |
| Configuring DNS Records for the Client Access Namespaces..... | 30 |
| Configuring Client Access Namespaces Using the Exchange Admin Center | 32 |
| Configuring Client Access Namespaces Using PowerShell | 34 |
| Exchange Server 2016 and SSL Certificates | 36 |
| SSL Certificate Requirements..... | 37 |
| Namespaces to Include on SSL Certificates | 38 |
| Which Type of Certificate to Purchase? | 39 |
| How Many SSL Certificates Should You Purchase?..... | 39 |
| Configuring the SSL Certificate | 40 |
| Testing the Client Access Configuration | 48 |
| What About External Access? | 48 |
| Configuring Transport..... | 50 |
| Inbound Mail Flow | 50 |
| Configuring Accepted Domains | 51 |
| Configuring Email Address Policies..... | 53 |
| Configuring MX Records in DNS..... | 55 |
| Configuring SMTP Connectivity to the Exchange Server | 57 |
| Testing Inbound Mail Flow..... | 58 |
| Outbound Mail Flow | 59 |
| Creating a Send Connector | 59 |
| Testing the Send Connector..... | 62 |
| SMTP Relay for Applications and Devices..... | 64 |
| Internal SMTP Relay..... | 65 |
| External SMTP Relay | 66 |

| | |
|---|----|
| External SMTP Relay Using Authentication | 67 |
| External SMTP Relay Using Anonymous Connections | 68 |
| Extra Considerations for SMTP Relay | 70 |
| Configuring Mailbox..... | 72 |
| Moving a Mailbox Database | 72 |
| Creating a New Mailbox Database..... | 73 |
| Configuring Mailbox Database Quotas | 74 |
| Managing Recipients..... | 77 |
| Creating Mailboxes | 77 |
| Mailbox-Enabling a User | 77 |
| Creating Multiple Mailboxes..... | 78 |
| Creating Distribution Groups | 79 |
| Managing Distribution Groups | 81 |
| Managing Delivery Restrictions for Distribution Groups..... | 82 |
| Using Moderation for Distribution Groups..... | 83 |
| Backup and Recovery | 84 |
| Backup and Recovery Terminology..... | 84 |
| Types of Backup | 84 |
| Backup Storage | 85 |
| Other General Terminology | 86 |
| What to Back Up for Exchange Server 2016 | 88 |
| Backing Up Exchange Server 2016 using Windows Server Backup | 89 |
| Restoring Mailbox Databases | 92 |
| Restoring Mailboxes and Items Using a Recovery Database..... | 97 |
| Creating a Recovery Database | 98 |
| Restoring a Mailbox Database into a Recovery Database | 98 |

| | |
|--|-----|
| Making a Restored Database Mountable | 103 |
| Running Mailbox Restore Requests | 105 |
| Managing Mailbox Restore Requests | 107 |
| Removing a Recovery Database..... | 108 |
| Recovering a Failed Exchange 2016 Server | 109 |
| Preparing the Server for Recovery..... | 110 |
| Performing a Recovery Install of Exchange 2016 | 110 |
| Restoring Custom Configurations | 110 |
| Restoring Databases After Server Recovery | 111 |
| What Next? | 112 |
| Appendix: Lab Setup Guide..... | 113 |
| Installing Hyper-V on Windows 8.1 or Windows 10 | 113 |
| Downloading Software | 114 |
| What Else Do You Need? | 115 |
| Building Virtual Machines in Hyper-V | 115 |
| Configuring a Virtual Switch..... | 116 |
| Creating a Virtual Hard Disk to Use as a Base Image..... | 117 |
| Creating a Virtual Machine to Use as a Base Image | 119 |
| Installing the Domain Controller..... | 131 |
| Install Certificate Services | 139 |
| Install a Virtual Machine for Exchange Server 2016 | 144 |
| Installing a Virtual Machine as a Client Machine..... | 146 |
| Lab Guide Summary | 146 |

Welcome

Hi there, and welcome the Exchange Server 2016 Quick Start Guide. This is just a quick introduction to let you know what to expect from this eBook.

With estimates of 20%-35% of Exchange mailboxes being in Office 365 one might wonder why should you bother learning about Exchange Server 2016. Of course, the simple fact is that even if 1/3rd of mailboxes are in Office 365, that still leaves 2/3rd of mailboxes hosted by on-premises Exchange servers.

And with Exchange Server 2016 being the very latest release of on-premises Exchange Server right now, that makes it well worth learning about today.

I wanted to write this eBook for one reason, to help you learn more about configuring and managing on-premises Exchange servers so that you can avoid simple errors that can have a big impact on your customers and end users.

In my consulting business I've worked with so many customers who were having problems with their on-premises Exchange due to some simple misconfigurations. It's not fair for anyone to declare that those customers should just move to the cloud and let Microsoft take care of running Exchange for them. I know it's not that simple, and that IT administrators all over the world just like you still need to manage and maintain on-premises Exchange, and will do so for many years ahead.

Here's what you can expect to learn by reading this eBook.

- **An introduction to Exchange Server 2016.** If you've got some experience with Exchange already then you'll learn about what's changed, and what's new. If you're just starting to work with Exchange, then you'll gain an understanding of the key concepts for Exchange Server 2016 (and most of them will help you with earlier versions of Exchange as well).
- **How to manage Exchange Server 2016.** I'll show you the management tools that you'll be using for Exchange, demonstrate how Roles Based Access Control helps you securely delegate administrative rights to your IT team, and then walk through a simple administrative task of creating a new user and mailbox.

- **Configuring Client Access.** You'll learn about the Client Access services for Exchange, how namespaces and SSL certificates come into play, and how to configure an Exchange server to allow end users to connect with Outlook, web browsers, and mobile devices.
- **Configuring Transport.** I'll show you how to establish inbound and outbound mail flow for your Exchange organization, as well as how to allow other applications and devices on your network to use Exchange for SMTP relay.
- **Configuring Mailbox.** You'll learn important database concepts, how to move databases to new storage locations, how to create new databases, and how to manage database settings such as mailbox quotas.
- **Managing Recipients.** You'll learn how to do daily administration tasks by creating users and distribution groups.
- **Backup and Recovery.** Become familiar with the very important task of protecting your Exchange data, and how to recover databases and mailboxes from backups.

In some places I will link you to an external resource, such as TechNet, to find out more. This is so you'll always be looking at the absolute latest information that is up to date and correct, rather than putting information in this eBook that may get outdated.

The absolute best way to become skilled in managing Exchange Server 2016 is to get hands on with a real server. But I don't want you messing with production servers that your customers and end users rely on. Instead, I recommend you create your own training environment to play around with.

To help you do that I've included a test lab setup guide in the appendix of this eBook. You can use the test lab guide to build your own training environment using Hyper-V on your Windows desktop or laptop, just like I use.

If you're going to build your own training environment I also strongly recommend that you purchase a domain name, if you don't already own one. This allows you to set up a realistic environment that can send and receive email with the outside world. Unfortunately, some ISPs block the SMTP ports for their customers' internet connections, so you may find that even with a domain name you have trouble with outbound mail flow. It's still worth buying one though, if you ask me.

If you don't have the resources to run your own test lab then don't worry, you'll still learn a lot just by reading this eBook. Now let's get started!

Introduction to Exchange Server 2016

For Exchange admins the release of a new on-premises version is an exciting time. And the release of Exchange Server 2016 is no different, there is certainly plenty to get excited about.

Exchange Server 2013 was the first version of Exchange “built for the cloud”. Microsoft learned a lot from running hosted, cloud-based Exchange services on Exchange Server 2007 and 2010-based servers. Back then it was called BPOS, and was the precursor to what we know as Office 365 today. Microsoft made a huge shift from their traditional software development methods, merging the cloud and on-premises product into one code base, which they delivered to on-premises customers as Exchange Server 2013.

While the initial “RTM” release of Exchange Server 2013 lacked polish, and was missing some key features, Microsoft was able to quickly deliver fixes and improvements through their quarterly “cumulative update” approach, and today Exchange Server 2013 stands as a solid and reliable on-premises version of Exchange.

In some ways Exchange Server 2016 is simply Exchange Server 2013 Service Pack 2. This isn’t intended to minimize the efforts of Microsoft’s Exchange product group, but stands more as an acknowledgement of the way Exchange Server 2016 builds on the quality foundation of Exchange Server 2013, and everything Microsoft has learned from running the product in Office 365 for millions of mailboxes.

Here’s what Microsoft themselves had to say about the [release of Exchange Server 2016](#):

“This version of Exchange is special because it was born in the cloud. From the depths of the mailbox store to the most visible parts of the Outlook web UI, the bits that make up Exchange 2016 are already in use across millions of mailboxes in Office 365.”

Key Features and Concepts

The key areas that Microsoft has focussed on with Exchange Server 2016 are:

- **Better collaboration** – with Outlook 2016 and the new “Outlook on the web” (the new name for OWA) document sharing will be easier, replacing traditional email attachments with links to OneDrive for Business or SharePoint 2016 (currently in Preview).
- **Improved Outlook web experience** – this is truly a big step forward for Outlook on the web which I personally use about 50% of the time. The quick action “Archive” button is my favourite, as well as the pasting of in-line images. Emojis are nice as well I suppose. The experience across different browsers and devices is optimized and more consistent as well.
- **Search** – faster, more flexible, more intelligent. Search can always be better of course.
- **Extensibility** – the add-in model for Outlook and Outlook on the web is in full swing. Interestingly the REST APIs have not made it into RTM, but we can likely expect to see those in a future update.
- **eDiscovery** – an important addition is the ability to search, hold and export public folder content. Microsoft has moved through the five stages of grief over public folders and is now in the acceptance stage. Public folders, once considered deprecated, will be around for a long time and need the same compliance features as mailboxes.
- **Simplified architecture** – combining Client Access and Mailbox services into a single server role greatly simplifies deployment and management. And the co-existence story for Exchange Server 2016 with Exchange 2013 and 2010 is set to make this one of the lowest friction upgrade paths in Exchange history.
- **High availability** – many performance and stability improvements that have flowed down to the on-premises product from Microsoft ongoing experience running Exchange Online.

The new features are certainly interesting, but what about features that didn’t make the cut? A number of items that have been publicly discussed in Microsoft blog posts and sessions at their Ignite conference are absent from Exchange 2016 RTM. Of course, all such information was subject to change before RTM. No doubt the primary driver here is to ensure features are fully developed and stable before shipping them in a future Cumulative Update for Exchange 2016.

A few of the missing features at RTM are:

- **Search index from passive** – the goal here is to have content indexes for passive database copies build/update from the passive database copy rather than replicate from

the active database copy, which should reduce DAG replication traffic. No timeline on when this feature will appear.

- **Auto-expanding archives** – the goal here is to have Exchange 2016 automatically provision additional archives for a user when their archive mailbox reaches 100Gb. This feature is still marked as “in development” on the [Office 365 roadmap](#), so you should expect to wait at least until it is rolling out in Office 365 before it will appear in an on-premises CU.
- **Delayed lag play down** – lagged copy play down will be enabled by default, causing lagged copies to automatically replay their log files and bring the database up to date if the DAG detects a loss of database redundancy, something that has reportedly avoided some potentially bad outage scenarios in Office 365. Delayed lag play down will throttle that replay process based on the server workload, ensuring it does not overload the server.

It may be disheartening to see key features not make it into the RTM build. But on the other hand, most Exchange Server RTM builds are missing something that many of us would consider important. The quarterly update cycle with features shipping in Cumulative Updates should deliver these key features to us in the near future. But they have to be stable first.

Server Roles Architecture

Exchange Server 2016 has just two server roles:

- **Mailbox server role** – this role will consolidate the Mailbox and Client Access roles from Exchange Server 2013. Compared to Exchange Server 2010 this role consolidates all of the functions of the Client Access, Mailbox, Hub Transport, and Unified Messaging server roles. The Mailbox server role in Exchange Server 2016 is the only mandatory server role, and the consolidation reinforces the recommended practice since Exchange Server 2010 to deploy Exchange as a multi-role server instead of deploying individual roles to separate servers.
- **Edge Transport server role** – this role will be much the same as Edge Transport in previous versions of Exchange, designed to sit in perimeter networks and provide secure inbound and outbound mail flow for the organization. Edge Transport servers are not mandatory.

What's New for Exchange Server 2013 Customers

For Exchange Server 2013 customers there are no unpleasant surprises and we can mostly enjoy the improvements in performance, manageability, and user experience, with no significant changes to the management tools and interfaces we're already used to.

The server roles architecture is a change from Exchange Server 2013, which had separate Client Access and Mailbox server roles. These are now consolidated into a single Mailbox server role for Exchange Server 2016.

Welcome news for Exchange Server 2013 customers is the capability for Exchange Server 2013 and 2016 to proxy client traffic to each other, instead of having to cut over all client access namespaces to the higher version during a co-existence period. This will ease the migration path for many customers.

Ultimately what you can expect from Exchange Server 2016 is improvements rather than wholesale changes.

What's New for Exchange Server 2010 Customers

For Exchange Server 2010 customers there's a new web-based administrative interface to learn, which is quite different from the MMC-based console that you would be used to already, but I assure you it is not difficult to master.

Public folder databases no longer exist, and have been replaced since Exchange Server 2013 with public folder mailboxes instead. For the end user there's no real changes here, but your management of public folders will change accordingly.

The server roles architecture is a more drastic change for Exchange Server 2010 customers who were used to the five server roles (Mailbox, Client Access, Hub Transport, Edge Transport, and Unified Messaging). Now you've got just two server roles to consider (Mailbox and Edge Transport), which is much simpler to deploy and manage.

The full list of changes since Exchange Server 2010 is quite long, but the important concepts around client access, transport, and mailbox remain very similar. You'll learn about these as you read through this eBook.

Design Fundamentals

These days Microsoft provides quite detailed guidance and tools for designing your Exchange Server 2016 deployments.

The Preferred Architecture

The Preferred Architecture is Microsoft's recommendation for deploying highly available and site resilient Exchange Server 2016 environments. The Preferred Architecture is based on Microsoft's experience running Office 365, with a key focus being on removing complexity and providing predictable failure responses.

Even if you do not plan to deploy a highly availability and site resilient Exchange Server 2016 environment the Preferred Architecture is still a worthwhile read. As this eBook is focussed on the fundamentals to get you started with Exchange Server 2016 I'll let you [read the Preferred Architecture](#) at your own leisure.

Sizing Guidance

Getting your server sizing right is critical to a successful Exchange Server 2016 deployment. If you undersize the server's processor, memory or storage you'll have unhappy users struggling with the poor server performance.

Every Exchange 2016 server you deploy should be sized according to [Microsoft's Exchange Server Role Requirements Calculator](#).

My recommendations for using the calculator are:

- Always download a new copy of the calculator before you begin. Microsoft often [release updates](#) that have important changes to sizing guidance that may impact your deployment.
- Take your time. It's a complicated tool and even the experts rarely get it right the first time.

- Read [Microsoft's sizing guidance](#) to understand the reasons why the calculator makes the recommendations that it does.

Virtualization and Storage

When you read the Preferred Architecture and various blog posts on the internet you would be forgiven for forming the view that you can't virtualize Exchange Server 2016. The fact is that it is supported to virtualize Exchange, and in my experience most customers do. However, if you are planning to deploy Exchange Server 2016 using virtualization make sure you follow the [virtualization guidelines from Microsoft](#) to stay within the supported boundaries.

Similarly, storage is an area of some confusion when it comes to Exchange Server 2016. A lot has been said about using cheap JBOD storage instead of expensive SAN. And it's true, Exchange is engineered to run very well on lower tier storage, and doesn't necessarily require SAN or even RAID for everything. However, as with virtualization make sure you read [Microsoft storage guidelines](#) for Exchange to ensure you stay within their recommendations and support boundaries.

System Requirements

Exchange Server 2016 has stricter requirements than previous versions of Exchange, however most of them are due to older versions of Windows Server falling out of support. For example, Windows Server 2003 has reached end of life, so in theory no customers should be running Windows Server 2003 in their environment any more. Accordingly, Microsoft has lifted the bar for their Exchange system requirements so that it is no longer supported to run on or alongside older, unsupported platforms.

Operating Systems

Exchange Server 2016 can be installed on:

- Windows Server 2012 Standard or Datacenter
- Windows Server 2012 R2 Standard or Datacenter

I strongly recommend you deploy on the latest supported version of Windows Server, which at this time is Windows Server 2012 R2.

Support for Windows Server 2016 is expected as well, but not until that operating system reaches RTM. Even then, you can expect that a specific cumulative update of Exchange Server 2016 will be required to run on Windows Server 2016, and that earlier builds such as Exchange Server 2016 RTM will remain unsupported.

Active Directory

Exchange Server 2016 has the following Active Directory requirements:

- Windows Server 2008 or higher domain controllers and global catalog servers
- Windows Server 2008 or higher domain and forest functional levels

It's fairly common for the domain and forest functional levels of Active Directory go untouched even when all of the legacy domain controllers have been upgraded or replaced. If you need to upgrade your functional levels ready through [Microsoft's guidance](#) first.

Exchange Organization

Exchange Server 2016 can be installed into an existing Exchange organization if it meets the following requirements:

- No Exchange Server 2007 or earlier versions of Exchange in the organization
- Any Exchange Server 2010 servers must be running at least [Service Pack 3 with Update Rollup 11](#)
- Any Exchange Server 2013 servers must be running at least [Cumulative Update 10](#)

You can have a combination of Exchange 2010 and 2013 servers in the organization as long as they each meet those minimum versions.

If you have Edge Transport servers or legacy server objects that were not properly removed from Active Directory you may encounter a setup error when you first try to install Exchange Server 2016. You can find more details about this, and the solutions, at the following article:

- [All Exchange 2013 Servers in the Organization Must Have Exchange 2013 Cumulative Update 10 or Later Installed](#)

Network and Firewall

Exchange Server 2016 supports IPv6, but only when IPv4 is also enabled on the server's network interfaces. It is not supported to disable IPv6 on your Exchange 2016 servers, even if you do not have an IPv6 network, except when Microsoft Support specifically instruct you to (and they will also ensure you disable it correctly). On an IPv6-capable network Exchange Server 2016 can use IPv6 to communicate.

The Windows Firewall on Exchange 2016 servers should be left enabled to help protect the server from network threats. The Exchange 2016 setup routine will automatically create Windows Firewall rules required for Exchange to operate. Although the Windows Firewall is recommended, it is not supported to use firewalls or other network devices to restrict the ports that are accessible between Exchange servers, or between Exchange servers and domain controllers. If you do have firewalls on the network between those servers an "any/any" rule is required. The exception to this rule is Edge Transport servers, which are normally placed in a perimeter network and therefore have clearly documented firewall requirements.

You can however use firewalls to restrict client access to Exchange Server 2016. Only a very small range of ports is required for client communications, which Microsoft has [published on TechNet](#).

Installing Exchange Server 2016

Now that we've gone over the basic system requirements for Exchange Server 2016 let's move on to installing an Exchange 2016 server. If you've built your training environment using the lab guide in Appendix A you can follow along with the installation as we go.

Which Edition of Exchange Server 2016 to Deploy?

For Exchange Server 2016 there are two editions of the server product itself, and there is only one difference between them which is the number of mounted databases per server.

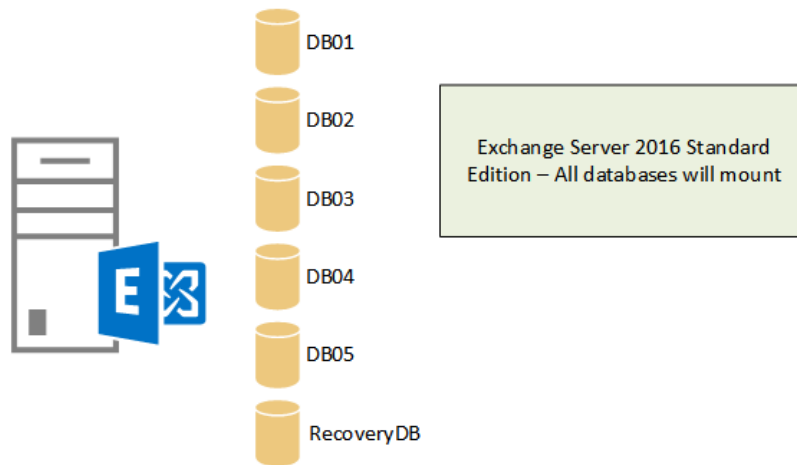
- Exchange Server 2016 Standard Edition – **maximum of 5** mounted databases per server
- Exchange Server 2016 Enterprise Edition – **maximum of 100** mounted databases per server

Microsoft's definition of a "mounted database" is:

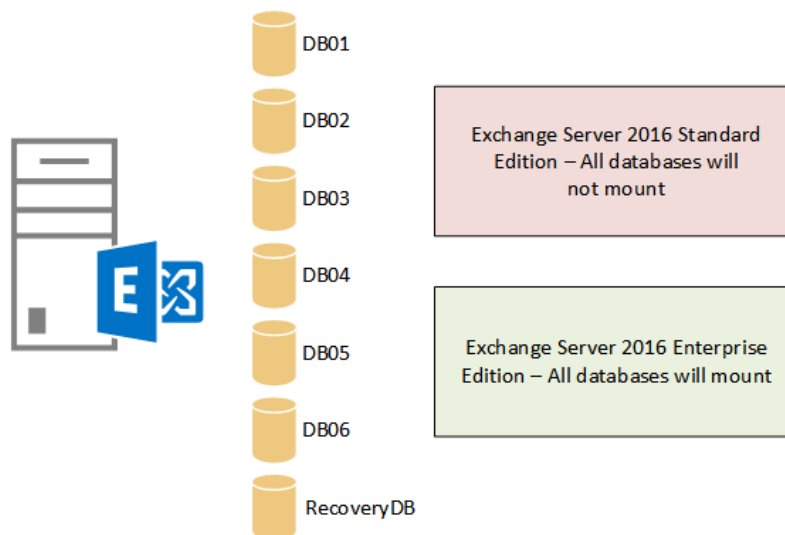
"A mounted database can be an active mailbox database that is mounted for use by clients, or a passive mailbox database that is mounted in recovery for log replication and replay. While you can create more databases than the limits described above, you can only mount the maximum number specified above. The recovery database does not count towards this limit."

Here's a few examples.

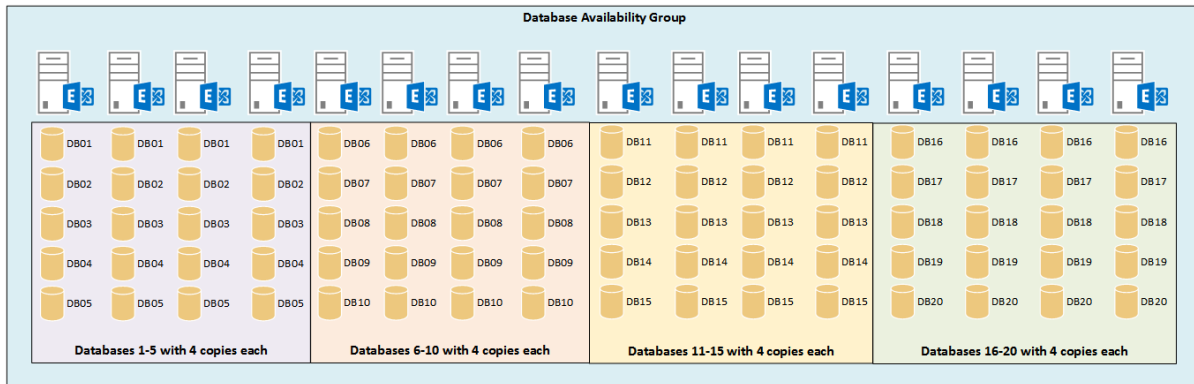
In this example a single Mailbox server running Standard Edition has 5 mailbox databases. All 5 databases will be able to mount, and an additional recovery database can also be created and mounted for any data restoration scenarios.



The same server running Standard Edition with 6 mailbox databases will not be able to mount all of the databases at the same time. However, if it is running Enterprise Edition it will be able to mount all 6 databases, or up to 100 databases.



What about a database availability group? DAGs can have up to 16 members, and each member is limited by the edition of Exchange Server 2016 that is installed. So a Standard Edition DAG member can host up to 5 active or passive database copies, and an Enterprise Edition DAG member can host up to 100 active or passive database copies. The DAG itself is only limited by the capabilities of all of its members. A DAG made up of 16 Standard Edition members, with each database having 4 copies, could therefore host up to 20 databases.



A DAG made up of 16 Enterprise Edition members, with each database having 4 copies, could host up to 400 databases.

To be clear, there is no requirement to run Exchange Server 2016 Enterprise Edition just because you're deploying a DAG. The choice of server edition is purely driven by the number of mounted databases each server will be hosting.

For the Edge Transport role, given it does not host any databases, it makes sense to use a Standard Edition server license.

When you purchase your Exchange Server 2016 server licenses you'll be provided with a license key that needs to be entered on the server. The license keys determines which server edition is installed, there is no different in installation media or installation method for each edition. All servers are first installed as a Trial Edition, and then you add your license key after installation is complete. You can upgrade from Trial to Standard, or from Trial to Enterprise. You can also upgrade from Standard to Enterprise. However, you can't downgrade from Enterprise to Standard without completely reinstalling the server. This means it is feasible to initially license your servers as Standard Edition, and then later upgrade them to Enterprise Edition if your environment scales up (e.g. if there is a corporate acquisition or merger).

As a final note, the information above applies only to the server licenses. The Client Access Licenses (CALs) are considered separately, and have no impact on the server license you choose to deploy and vice versa. CALs determine which features a given mailbox user can make use of. You can read more about the usage rights of each CAL on Microsoft's [Exchange Server 2016 licensing page](#).

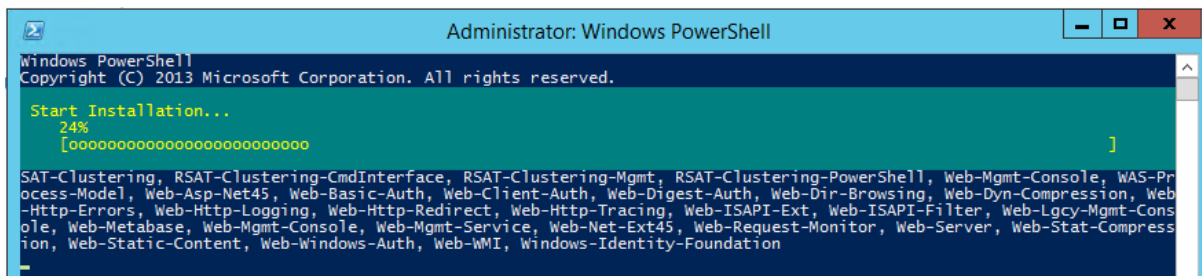
Installing Exchange Server 2016 Pre-Requisites on Windows Server 2012 R2

[Exchange Server 2016](#) can be installed on Windows Server 2012 and Windows Server 2012 R2. For both versions of Windows Server either the Standard or Datacenter edition can be used to run Exchange Server 2016. Exchange itself does not rely on any specific features of either the Standard or Datacenter editions.

Note that a full server installation with GUI is required for Exchange Server 2016, it can't be installed on a Core mode installation of Windows Server.

For an Exchange Server 2016 Mailbox server installation open an elevated (run as administrator) PowerShell console and run the following command to install the operating system roles and features.

```
C:\> Install-WindowsFeature AS-HTTP-Activation, Desktop-Experience, NET-Framework-45-Features, RPC-over-HTTP-proxy, RSAT-Clustering, RSAT-Clustering-CmdInterface, RSAT-Clustering-Mgmt, RSAT-Clustering-PowerShell, Web-Mgmt-Console, WAS-Process-Model, Web-Asp-Net45, Web-Basic-Auth, Web-Client-Auth, Web-Digest-Auth, Web-Dir-Browsing, Web-Dyn-Compression, Web-Http-Errors, Web-Http-Logging, Web-Http-Redirect, Web-Http-Tracing, Web-ISAPI-Ext, Web-ISAPI-Filter, Web-Lgcy-Mgmt-Console, Web-Metabase, Web-Mgmt-Console, Web-Mgmt-Service, Web-Net-Ext45, Web-Request-Monitor, Web-Server, Web-Stat-Compression, Web-Static-Content, Web-Windows-Auth, Web-WMI, Windows-Identity-Foundation
```



A restart is required after the roles and features have finished installing. If you'd prefer that the server restarts itself automatically simply append **-Restart** to the command.

After the restart download and install (in order):

- [.NET Framework 4.5.2](#)
- [Microsoft Unified Communications Managed API 4.0, Core Runtime 64-bit](#)

The server is now ready to install Exchange Server 2016.

Running Exchange Server 2016 Setup

After you've prepared a Windows Server with the Exchange Server 2016 pre-requisites you can proceed with the installation of Exchange Server itself.

- Before you start there are a few things to be aware of:
- Installing Exchange Server 2016 requires an Active Directory schema update. We'll look at that in more detail shortly.
- Aside from the schema update installing Exchange Server 2016 makes other irreversible changes to your Active Directory forest. If you've never backed up your Active Directory, or you've never heard of a forest recovery, [here's some reading for you](#).
- If you're installing Exchange into the forest for the first time you will be choosing an organization name. The Exchange organization can't be renamed at a later date, so choose a name you're happy with keeping forever.

A new installation of Exchange Server 2016 involves applying an Active Directory schema update, as do most Exchange Server cumulative updates, as well as preparing the Active Directory domains where Exchange Server 2016 and any mail-enabled objects will be located. In an Active Directory forest with a single domain this can all be performed as one task.

The Active Directory schema update will automatically apply when you run Exchange Server 2016 setup on the first server in your environment. A Windows Server 2012 R2 server with the Exchange Server 2016 Mailbox server role pre-requisites installed doesn't quite meet the requirements (you'll need to add the RSAT-ADDS feature as shown below). A domain controller will have RSAT-ADDS installed already, but may also need the .NET Framework version shown below to be installed first.

Whether you're running the schema update from an Exchange server or a separate server (some organizations do it as a separate task due to change control reasons, or because of different teams having different administrative responsibilities in the environment) then the following requirements apply:

- [.NET Framework 4.5.2](#) must be installed
- The RSAT-ADDS feature must be installed

```
C:\> Install-WindowsFeature RSAT-ADDS
```

- The forest functional level must be at least Windows Server 2008
- The account used to run the schema update and Active Directory preparation must be a member of Enterprise Admins and Schema Admins. These are high privilege groups I recommend you plan to remove your account from the groups when you're done with this task. Note, if you've just added yourself to these groups you'll need to log out and back in to the server for the new group membership to take effect.
- The server you're running the schema update from must be located in the same Active Directory site as the Schema Master. You can identify your Schema Master by running my [Get-ADInfo.ps1](#) script, or by using the Get-ADForest PowerShell cmdlet.

```
PS C:\> (Get-ADForest).SchemaMaster
```

Now we're ready to run the Active Directory schema update and preparation.

If you've already got Exchange Server running in your environment you can check the current Exchange schema version before applying the update, so that you can see what the before and after version numbers are.

In PowerShell run the following one-liner created by [Exchange Server MVP Michael B Smith](#):

```
PS C:\> "Exchange Schema Version = " + ([ADSI]("LDAP://CN=ms-Exch-Schema-Version-Pt," +
([ADSI]"LDAP://RootDSE").schemaNamingContext)).rangeUpper
Exchange Schema Version =
```

Note, in my example above there is no existing Exchange server installed, hence no Exchange schema version to report.

Extract the Exchange Server 2016 setup files into a folder, open a command prompt window, and then navigate to the location where the Exchange setup files were extracted.

To apply only the schema update run the following command:

```
C:\Admin\ex2016>setup /PrepareSchema /IAcceptExchangeServerLicenseTerms

Welcome to Microsoft Exchange Server 2016 Unattended Setup
Copying Files...
File copy complete.
Setup will now collect additional information needed for
installation.

Performing Microsoft Exchange Server Prerequisite Check

    Prerequisite Analysis                                COMPLETED

Configuring Microsoft Exchange Server
```

Extending Active Directory schema

COMPLETED

The Exchange Server setup operation completed successfully.

After applying the schema update we can check the version number again.

```
PS C:\> "Exchange Schema Version = " + ([ADSI]("LDAP://CN=ms-Exch-Schema-Version-Pt," +  
([ADSI]"LDAP://RootDSE").schemaNamingContext)).rangeUpper  
Exchange Schema Version = 15317
```

To prepare Active Directory run one of the following commands. Note this will also apply the schema update if you did not perform that step already.

If you do not already have an Exchange organization you'll need to provide a name for the organization now, for example:

```
C:\Admin\ex2016>setup /PrepareAD /OrganizationName:"Exchange Lab"  
/IAcceptExchangeServerLicenseTerms
```

If you're installing Exchange Server 2016 into an existing Exchange organization you do not need to specify the organization name, for example:

```
C:\Admin\ex2016>setup /PrepareAD /IAcceptExchangeServerLicenseTerms
```

Remember, you can't change the Exchange organization name later, so choose a name you'll be happy to live with forever. Also, after installing Exchange Server 2016 as a new organization you will not be able to install any earlier versions of Exchange into the same organization.

```
C:\Admin\ex2016>setup /PrepareAD /OrganizationName:"Exchange Lab" /IAcceptExchan  
geServerLicenseTerms
```

```
Welcome to Microsoft Exchange Server 2016 Unattended Setup  
Copying Files...  
File copy complete.  
Setup will now collect additional information needed for  
installation.
```

Performing Microsoft Exchange Server Prerequisite Check

Prerequisite Analysis

COMPLETED

Setup will prepare the organization for Exchange Server 2016 by using 'Setup /PrepareAD'. No Exchange Server 2007 roles have been detected in this topology. After this operation, you will not be able to install any Exchange Server 2007 roles.

For more information, visit: [http://technet.microsoft.com/library\(EXCHG.150\)/ms.exch.setupreadiness.NoE12ServerWarning.aspx](http://technet.microsoft.com/library(EXCHG.150)/ms.exch.setupreadiness.NoE12ServerWarning.aspx)

Setup will prepare the organization for Exchange Server 2016 by using 'Setup /PrepareAD'. No Exchange Server 2010 roles have been detected in this topology. After this operation, you will not be able to install any Exchange Server 2010 roles.

For more information, visit: [http://technet.microsoft.com/library\(EXCHG.150\)/ms.exch.setupreadiness.NoE14ServerWarning.aspx](http://technet.microsoft.com/library(EXCHG.150)/ms.exch.setupreadiness.NoE14ServerWarning.aspx)

Configuring Microsoft Exchange Server

Organization Preparation

COMPLETED

The Exchange Server setup operation completed successfully.

If you have additional domains in your forest that you need to prepare (any domain that will host an Exchange server or mail-enabled objects) follow the [guidance on TechNet](#).

The Mailbox server role contains all of the components required to run an Exchange Server 2016 server. After installing the Exchange Server 2016 pre-requisites on a server you can install the Exchange Server 2016 Mailbox server role by running the following command from an elevated command prompt.

```
C:\Admin\ex2016>setup /Mode:Install /Role:Mailbox /IAcceptExchangeServerLicenseTerms
```

```
Welcome to Microsoft Exchange Server 2016 Unattended Setup
Copying Files...
File copy complete.
Setup will now collect additional information needed for
installation.
```

```
Languages
Management tools
Mailbox role: Transport service
Mailbox role: Client Access service
Mailbox role: Unified Messaging service
Mailbox role: Mailbox service
Mailbox role: Front End Transport service
Mailbox role: Client Access Front End service
```

Performing Microsoft Exchange Server Prerequisite Check

| | |
|---------------------------|-----------|
| Configuring Prerequisites | COMPLETED |
| Prerequisite Analysis | COMPLETED |

Configuring Microsoft Exchange Server

| | |
|---|-----------|
| Preparing Setup | COMPLETED |
| Stopping Services | COMPLETED |
| Copying Exchange Files | COMPLETED |
| Language Files | COMPLETED |
| Restoring Services | COMPLETED |
| Language Configuration | COMPLETED |
| Exchange Management Tools | COMPLETED |
| Mailbox role: Transport service | COMPLETED |
| Mailbox role: Client Access service | COMPLETED |
| Mailbox role: Unified Messaging service | COMPLETED |
| Mailbox role: Mailbox service | COMPLETED |
| Mailbox role: Front End Transport service | COMPLETED |
| Mailbox role: Client Access Front End service | COMPLETED |
| Finalizing Setup | COMPLETED |

The Exchange Server setup operation completed successfully.
Setup has made changes to operating system settings that require a reboot to take effect. Please reboot this server prior to placing it into production.

After setup has completed restart the server before you continue with configuring Exchange Server 2016.

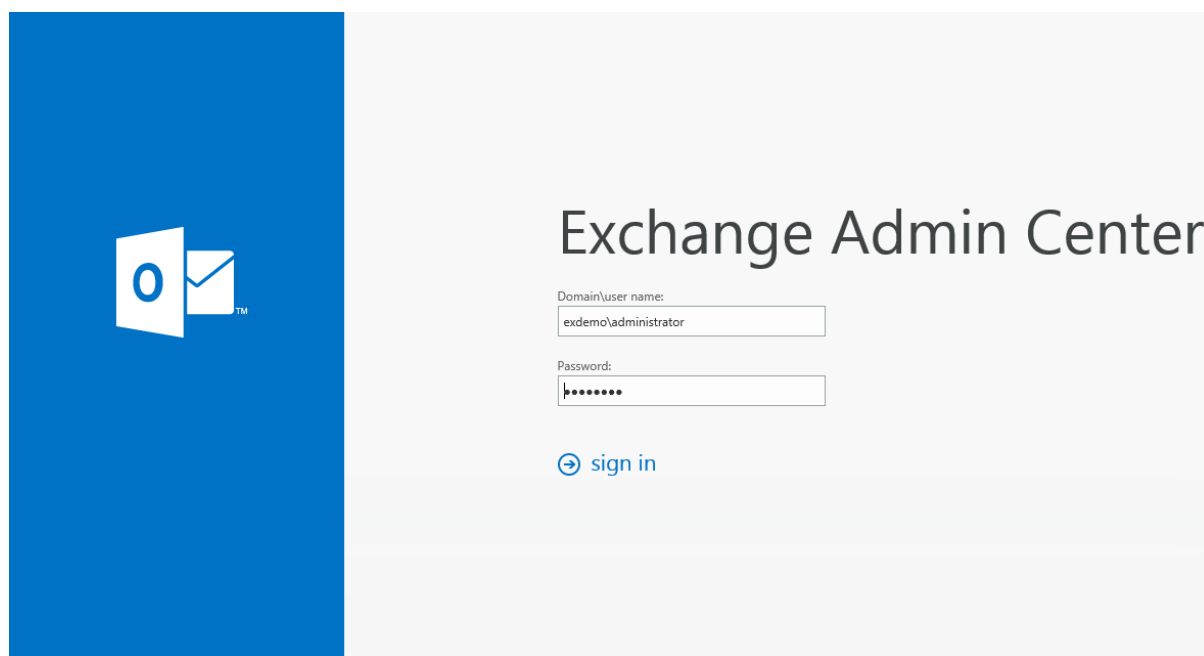
Managing Exchange Server 2016

Before we start to walk through the configuration of Exchange Server 2016 let's first take a look at the management tools and concepts that we'll be dealing with.

Introduction to Management Tools

There are two management interfaces that you'll use most of the time to manage your Exchange servers. These are the management tools for Exchange Server 2016 itself. In addition to the Exchange management tools you'll also need to use management tools for Active Directory, DNS, Windows Server, and many others. You've probably already got some experience with those other tools, but even if you don't I'll show you any specific tasks that need to be performed in them for Exchange as we go through this eBook.

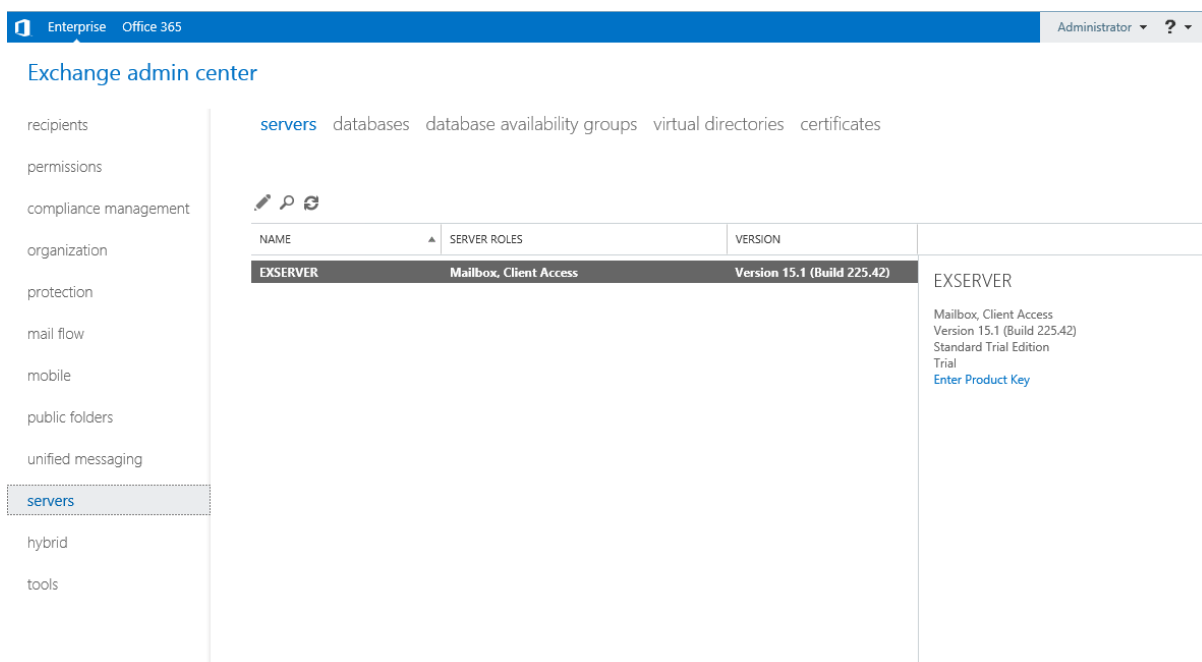
The Exchange Admin Center (EAC) is the web-based administrative interface for Exchange Server 2016. You can access the EAC on a server by opening a web browser and going to <https://servername/ecp>.



The /ecp is for “Exchange Control Panel” which is the web-based control panel that allows users to access options such as their out of office settings using only a web browser. The EAC interface for administrators is delivered from the same /ecp virtual directory.

When you access the server’s EAC URL you will probably see an SSL certificate warning in your web browser. At this stage you can ignore it, and we’ll be fixing it later so that the correct URL and SSL certificate can be used.

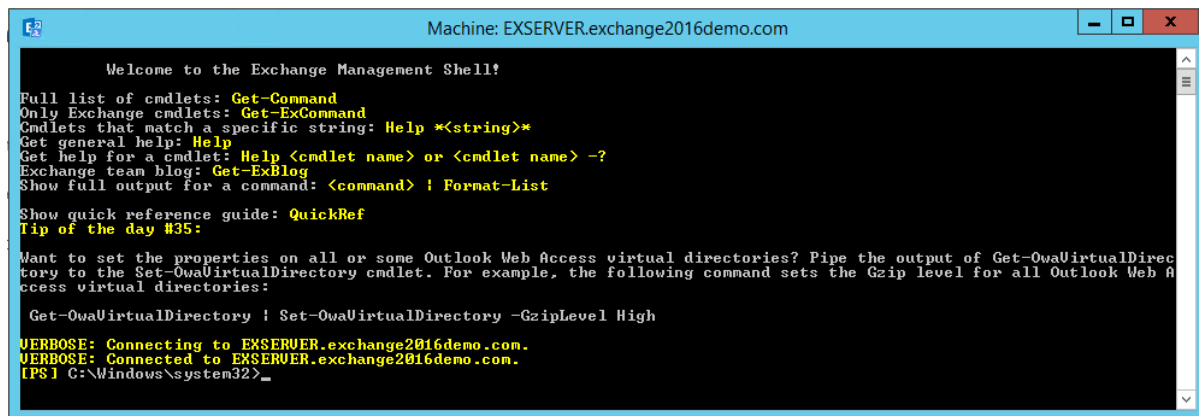
After you’ve logged in to the EAC with an administrative account you’ll see a list of categories down the left hand side, and in each one a series of sections across the top of the page. The middle section of the page will display lists of objects for that section, with controls for searching, editing, and other options depending on which section you’re in. On the right hand side there will be a list of actions that you can perform, again depending on which section you’re in and the object that you’ve selected.



You can also administer Exchange Server 2016 using the Exchange Management Shell (EMS). You’ll find the EMS in the list of apps on the server, and it can also be installed on Windows 8.1 computers.

The EMS is PowerShell with an Exchange management module loaded and a connection established to an Exchange server in your organization. Everything that you can perform in the

EAC can be performed in the EMS as well, but the EMS permits many more administrative tasks to be performed, and is also better for bulk administration and automation.



```
Machine: EXSERVER.exchange2016demo.com

Welcome to the Exchange Management Shell!

Full list of cmdlets: Get-Command
Only Exchange cmdlets: Get-ExCommand
Cmdlets that match a specific string: Help *(<string>)*
Get general help: Help
Get help for a cmdlet: Help <cmdlet name> or <cmdlet name> -?
Exchange team blog: Get-ExBlog
Show full output for a command: <command> ! Format-List

Show quick reference guide: QuickRef
Tip of the day #35:

Want to set the properties on all or some Outlook Web Access virtual directories? Pipe the output of Get-OwaVirtualDirectory to the Set-OwaVirtualDirectory cmdlet. For example, the following command sets the Gzip level for all Outlook Web Access virtual directories:

Get-OwaVirtualDirectory | Set-OwaVirtualDirectory -GzipLevel High

VERBOSE: Connecting to EXSERVER.exchange2016demo.com.
VERBOSE: Connected to EXSERVER.exchange2016demo.com.
[PS] C:\Windows\system32>
```

Many of the configuration examples throughout this eBook will involve running commands in the Exchange Management Shell.

Role-Based Access Control

Exchange Server 2016 uses an administrative permissions model that is separate from Active Directory. The account that you used to install Exchange will be granted what is called “Organization Management” rights by default, which allows the account to do almost anything in the Exchange organization. What the Organization Management group can’t do by default can be granted by someone in the Organization Management group anyway, so there are no security barriers standing between an org admin and doing whatever they want to the Exchange environment.

Obviously such a powerful administrative privilege should not be handed out to every administrator in the IT team, which is why several other pre-configured role groups also exist. You can read the full set of role groups on [TechNet](#), but I want to call out a few of the key roles that will be useful for most IT departments:

- Organization Management – members of this role group can administer the entire Exchange organization. We’ll be using an account in this role group as we work through the configuration of Exchange Server 2016 in this eBook, but for day to day administrative tasks you most likely will not need this level of access.
- View-Only Organization Management – members of this role group can view but not modify everything in the organization. Effectively this is a “look but don’t touch” role.

This is a useful role group for service accounts that are used to run reporting scripts, so that they can retrieve all the information they need but without the risk of accidentally modifying something.

- Recipient Management – this role group is a good general purpose, day to day administration account for Help Desk and lower level support staff. I also use this group for my own general administration, and keep a separate administrative account with Organization Management access that I only use when those higher permissions are required.

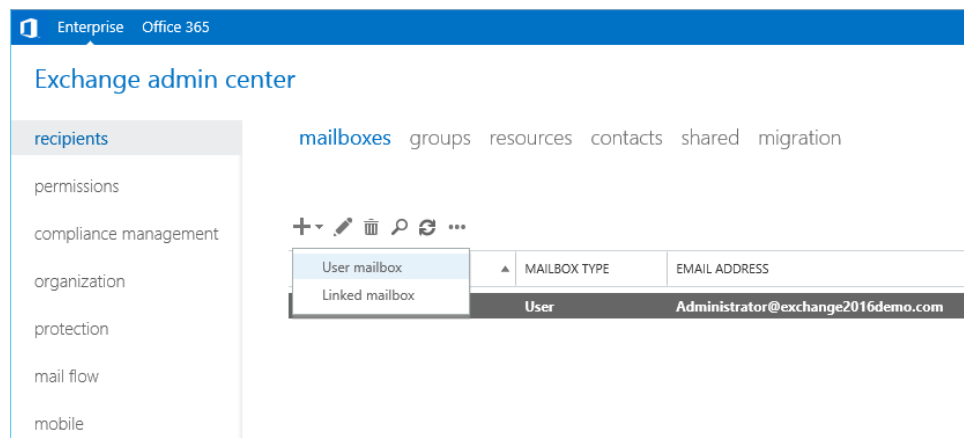
What State is the Exchange 2016 Server in Right Now?

So far we've installed Exchange Server 2016 into a new environment. But what state is the server currently in? Can it be used to access mailboxes, or to send and receive email? Let's take a look at what is working and what is not working, so that you can understand why the configuration performed throughout this eBook is actually necessary.

Creating a User and Mailbox

To demonstrate the state of the server let's create a user account and enable a mailbox for that user account. Log into the EAC and navigate to **Recipients** → **Mailboxes**.

Create a new **User mailbox**.



Fill out the new user form. In this example I'm creating a mailbox for a **New user**, which will create the Active Directory user object as well. As such, I need to fill out details such as the

name, organizational unit, and password. If you already have an Active Directory user then choose **Existing user** instead. Click **Save** when you're finished.

new user mailbox

Alias:
Adam.Wally

☐ Existing user
Browse...

☒ New user

First name:
Adam

Initials:

Last name:
Wally

*Display name:
Adam Wally

*Name:
Adam Wally

Organizational unit:
exchange2016demo.com, X Browse...

*User logon name:
adam.wallyv @ exchange2016demo.cc

Save Cancel

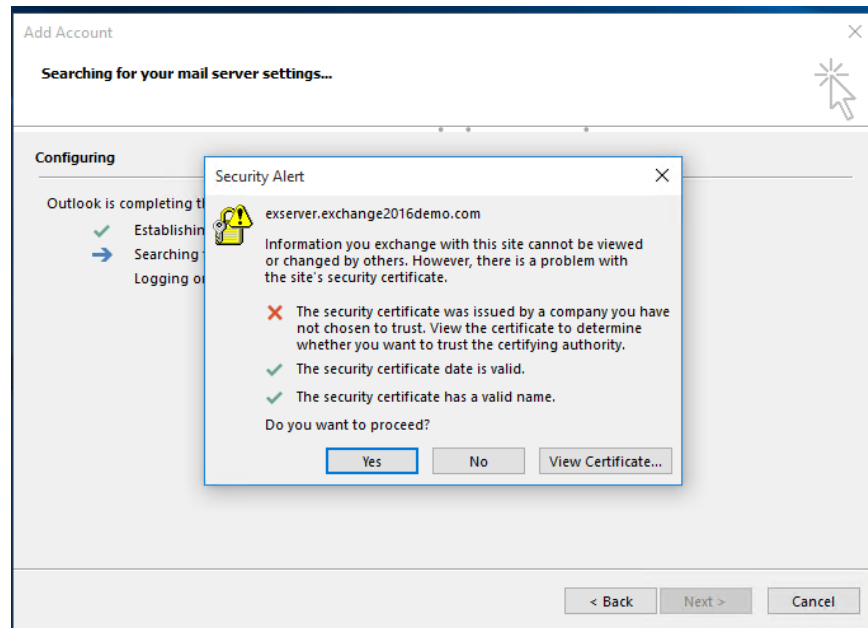
The user's alias is the portion of the email address on the left side of the @ symbol. It must be unique in your organization.

If you were curious enough to click on the **More options** link at the bottom of the form you'll see additional choices for the mailbox databases, whether archive-enable the mailbox, and the address book policy. None of these are mandatory. In fact, Exchange will automatically choose an available mailbox database to place the mailbox on if you do not specify one.

Accessing the New User's Mailbox

Now that we've got a new user and mailbox in the environment let's take a look at the state of the Exchange server. The simplest way to demonstrate this is to try to use Outlook to connect to a mailbox, send email, or receive email.

On a Windows 10 computer I've logged on with the user Adam Wally that I created earlier, and launched Outlook 2016. While stepping through the initial configuration process a security alert appears.



The two most common problems reported by the Outlook certificate warning message are:

- The name on the security certificate is invalid or does not match the name of the site
- The security certificate was issued by a company you have not chosen to trust

When you install Exchange Server 2016 into your Active Directory environment the setup process registers a Service Connection Point (SCP) for the Autodiscover service. Autodiscover is used by client applications to discover information about Exchange mailboxes and services. For example, Outlook uses Autodiscover during the setup of a new Outlook profile to discover the server settings for the user, so that the profile can be automatically configured (instead of the old days of manually entering server names and other details into Outlook).

By default, the Autodiscover SCP is registered using a URL that includes the Exchange server's fully-qualified domain name. You can see the Autodiscover URL for an Exchange 2016 server by running the **Get-ClientAccessService** cmdlet in the Exchange Management Shell.

For example:

```
[PS] C:\>Get-ClientAccessService -Identity EXSERVER | Select AutodiscoverServiceInternalUri
AutodiscoverServiceInternalUri
-----
https://exserver.exchange2016demo.com/Autodiscover/Autodiscover.xml
```

Autodiscover is accessible via an HTTPS (SSL) connection from clients. The Exchange server also has a number of other web services that are accessible using HTTPS connections from clients,

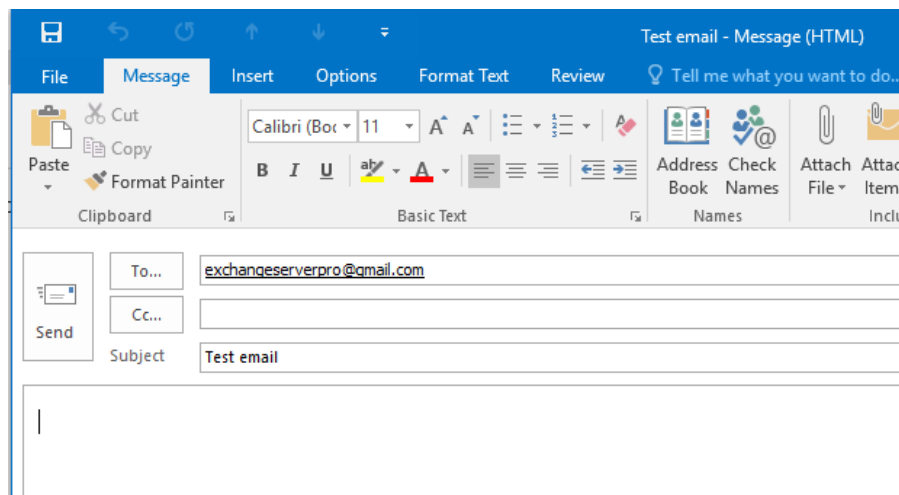
such as Exchange Web Services (EWS), Outlook on the web (also known as OWA), ActiveSync (for mobile devices), and Outlook Anywhere (used by Outlook clients).

As the connection is over HTTPS the SSL certificate configured on the server must meet three criteria to be considered valid by the client:

- The certificate was issued by a trusted certificate authority (CA)
- The certificate has not expired
- The name on the certificate matches the server name (or URL) that the client is connecting to

In the next chapter of this eBook we'll configure the Exchange namespaces and SSL certificate so that this error no longer appears.

But first let's look at what else doesn't work yet. We can ignore the certificate warning in Outlook and proceed with the profile configuration to open the mailbox. The next item to test is whether the user can send or receive email.



The test email above is sent to a Gmail recipient. It won't arrive, and can be seen stuck in the queue of the Exchange server.

```
[PS] C:\>Get-Queue
```

| Identity | DeliveryType | Status | MessageCount | Velocity | RiskLevel | OutboundIPPool |
|----------------------|--------------|--------|--------------|----------|-----------|----------------|
| EXSERVER\Submission | Undefined | Ready | 0 | 0 | Normal | 0 |
| EXSERVER\Unreachable | Unreachable | Ready | 1 | 0 | Normal | 0 |

Outbound email relies on a Send Connector being configured to route the email out of the organization to other email servers on the internet. If you happen to be deploying Exchange

Server 2016 into an existing organization that has other servers and Send Connectors already in place, then outbound email from Exchange 2016 mailboxes will likely work right away. For this example scenario, with the server being deployed into a new environment, no Send Connectors exist yet.

Similarly, inbound email to the recipient adam.wally@exchange2016demo.com will not work without the MX records being added to DNS, and the correct firewall points being opened to allow the inbound SMTP connections to be made.

We'll cover both inbound and outbound mail flow in the next chapter of this eBook. For now we've clearly established that the newly installed Exchange 2016 server still has some important configurations to be performed.

Configuring Client Access

Although the server role architecture in Exchange Server 2016 removed the “Client Access server” as a role, there still exists Client Access services in a logical sense. In this chapter we’ll look at configuring the Client Access services for the newly installed Exchange 2016 server.

When you first install Exchange Server 2016 it is pre-configured with default URLs for the various HTTPS services such as OWA (Outlook on the web, formerly known as Outlook Web App), ActiveSync (mobile device access), Exchange Web Services (the API used for a variety of client communications), and others.

The default URLs contain the fully qualified domain name of the server. So for example if your server name is “**exchange01.domain.com**” then the default URL for OWA will be “**https://exchange01.domain.com/owa**”.

These default URLs allow the services to function but they are not suitable for production deployments for several reasons, such as:

- They are difficult for end users to remember (this primarily impacts Outlook on the web, where users tend to find it easier to remember a URL such as “**webmail.domain.com**”)
- A URL containing a specific server name can’t be load-balanced across multiple servers in a high availability deployment
- The internal AD namespace for many organizations is not a valid domain name on the internet, for example domain.local, which makes it impossible to acquire SSL certificates for Exchange 2016 (I’ll cover SSL certificates in a separate article coming soon)

The recommended practice is to change the URLs configured on your Exchange 2016 servers to aliases or generic host names such as “**mail.domain.com**” after you first install the server.

While there are a variety of namespace designs that apply to different deployment scenarios I will demonstrate here the simplest approach, which is to configure the same namespace (URL) for all services. I’ll be demonstrating with a single Exchange Server 2016 server, but this approach can also be used if you have multiple Exchange servers that you want to load balance.

In my example scenario:

- The server’s real name is **exserver.exchange2016demo.com**

- The namespace I'll be using is mail.exchange2016demo.com
- Internal and external namespaces will be the same

Using my [GetExchangeURLs.ps1](#) script I can see the current configuration of the server.

```
PS C:\Scripts> .\GetExchangeURLs.ps1 -Server EXSERVER

-----
Querying EXSERVER
-----

Outlook Anywhere
- Internal: exserver.exchange2016demo.com
- External: exserver.exchange2016demo.com

Outlook Web App
- Internal: https://exserver.exchange2016demo.com/owa
- External:

Exchange Control Panel
- Internal: https://exserver.exchange2016demo.com/ecp
- External:

Offline Address Book
- Internal: https://exserver.exchange2016demo.com/OAB
- External:

Exchange Web Services
- Internal: https://exserver.exchange2016demo.com/EWS/Exchange.asmx
- External:

MAPI
- Internal: https://exserver.exchange2016demo.com/mapi
- External:

ActiveSync
- Internal: https://exserver.exchange2016demo.com/Microsoft-Server-ActiveSync
- External:

Autodiscover
- Internal SCP: https://exserver.exchange2016demo.com/Autodiscover/Autodiscover.xml

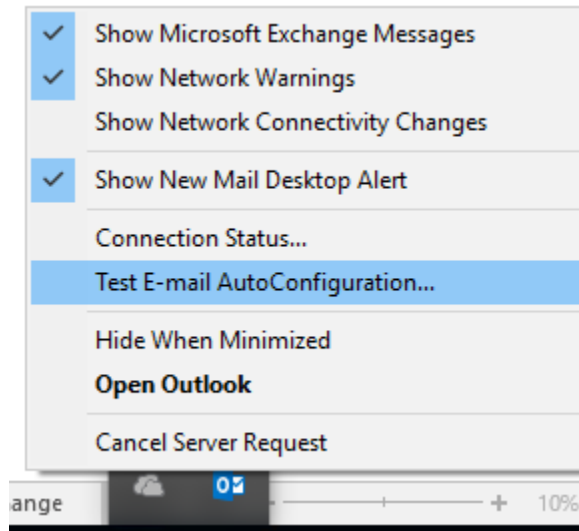
Finished querying all servers specified.
```

That's the server-side view of things, but what does this mean for clients that are trying to connect to Exchange?

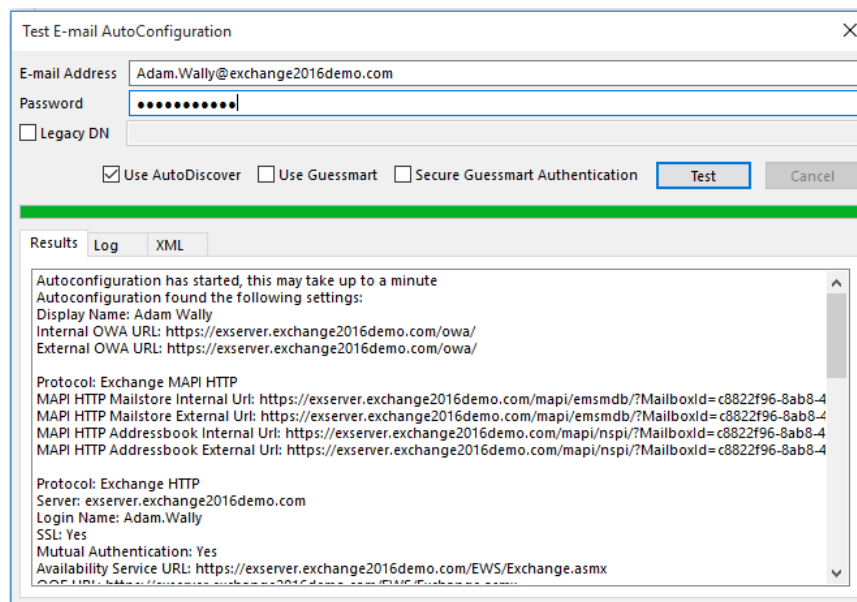
Clients such as Outlook and mobile devices use the Autodiscover service to find out which URLs they should connect to for different Exchange services. For example, when you launch Outlook for the first time it looks up the Autodiscover service in Active Directory, which is published as a Service Connection Point (SCP) when Exchange is first installed (this is also referred to as the AutoDiscoverInternalURI).

A query is sent to the Autodiscover service to learn which URLs Outlook should connect to for accessing that particular user's mailbox. Autodiscover returns an XML response with that

information in it. You can see a view of what Outlook receives back from Autodiscover by running the **Test E-mail AutoConfiguration** test. To start the test, hold the CTRL key and right-click the Outlook icon in your system tray.



Enter the password for the user account, and clear the two Guessmart tick boxes before you launch the test. In the results you'll see a list of URLs for services such as OWA, MAPI HTTP, OAB, and others.



As we learned earlier those URLs that contain the server's real name are not suitable and need to be changed.

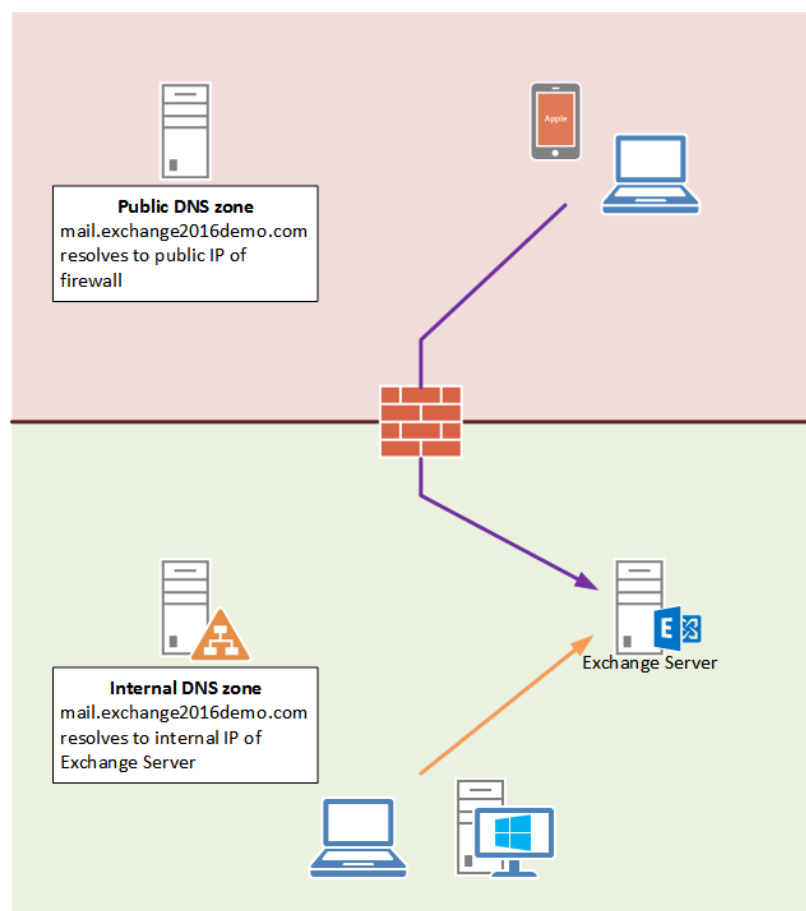
Configuring DNS Records for the Client Access

Namespaces

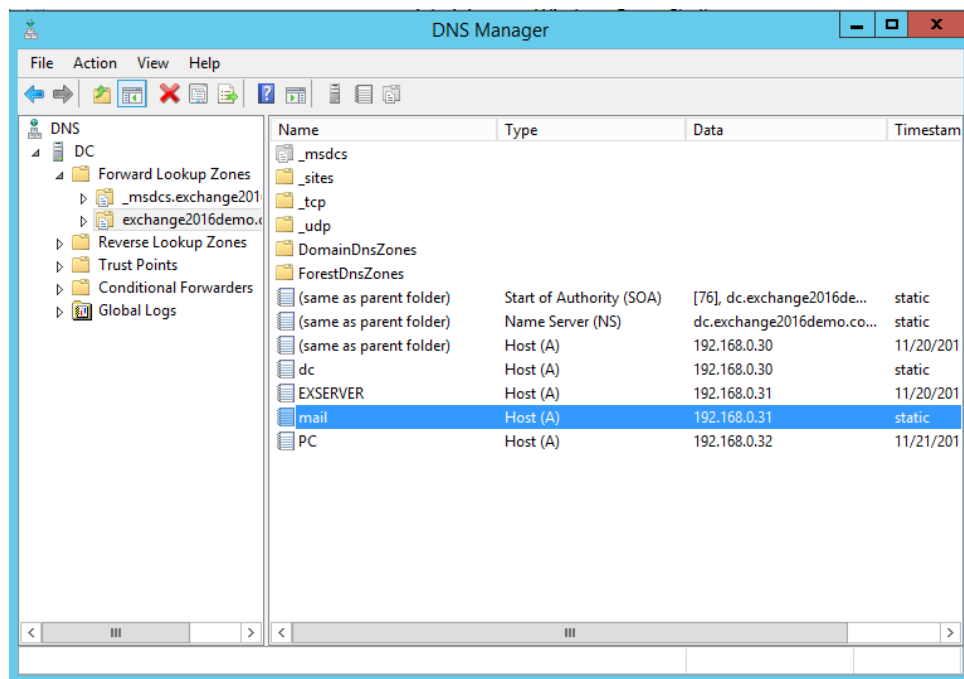
Before changing your server's namespace configuration you should make sure that the DNS records for the new namespaces already exist in DNS. Some of the virtual directory configuration tasks can fail if the name you specify isn't resolvable in DNS.

In this example scenario I'll be using split DNS, which is a recommended practice for Exchange Server 2016 deployments. Split DNS means I will host a DNS zone on my internal DNS servers, and use that to resolve **mail.exchange2016demo.com** to the internal IP address of my Exchange server (or load balancer if this was a high availability deployment).

Meanwhile, the public DNS zone also has a **mail.exchange2016demo.com** record that resolves to the public IP address of my firewall or router, which will then NAT any external connections to the Exchange server's internal IP.



Your internal DNS zone is usually hosted by the Active Directory domain controllers in your environment, and managed using the DNS management console.



Your public DNS zone is usually hosted by a DNS hosting provider, often the same company that was used to register the domain name. The DNS hosting provider will have a self-service portal you can use to manage your DNS records.

Add the records to both of the zones in your split DNS configuration and make sure they are resolving correctly before you continue. Here I am using PowerShell to query the internal DNS servers used by my client machine to verify the internal record, and then querying the Google DNS servers to verify the external record.

```
PS C:\> Resolve-DnsName mail.exchange2016demo.com
```

| Name | Type | TTL | Section | IPAddress |
|---------------------------|------|------|---------|---------------|
| mail.exchange2016demo.com | A | 3600 | Answer | 192.168.0.126 |

```
PS C:\> Resolve-DnsName mail.exchange2016demo.com
```





| Name | Type | TTL | Section | IPAddress |
|---------------------------|------|------|---------|-----------------|
| mail.exchange2016demo.com | A | 3600 | Answer | 203.206.161.219 |

Configuring Client Access Namespaces Using the Exchange Admin Center

After logging in to the Exchange Admin Center in your organization navigate to **Servers** > **Virtual Directories** and select the server you want to configure. There are two approaches you can take. The first is clicking the wrench icon to configure the external namespace for one or more servers.

servers databases database availability groups **virtual directories** certificates

Select server: All servers
Select type: All

| NAME | SERVER | TYPE | VERSION | LAST MODIFIED T... |
|-------------------------------|----------|---------|---------------------------|--------------------|
| Autodiscover (Default W... | EXSERVER | Auto... | Version 15.1 (Build ... | 13/10/2015 9:2... |
| ecp (Default Web Site) | EXSERVER | ECP | Version 15.1 (Build 22... | 21/11/2015 1:40... |
| EWS (Default Web Site) | EXSERVER | EWS | Version 15.1 (Build 22... | 21/11/2015 1:40... |
| Microsoft-Server-ActiveSyn... | EXSERVER | EAS | Version 15.1 (Build 22... | 21/11/2015 1:40... |
| OAB (Default Web Site) | EXSERVER | OAB | Version 15.1 (Build 22... | 21/11/2015 1:40... |
| owa (Default Web Site) | EXSERVER | OWA | Version 15.1 (Build 22... | 21/11/2015 1:39... |
| PowerShell (Default Web Si... | EXSERVER | Powe... | Version 15.1 (Build 22... | 13/10/2015 9:26... |

Autodiscover
Authentication: Bz
Point Security, OA

A window appears that allows you to add one or more servers and specify an external namespace to use.

configure external access domain

Select the Client Access servers to use with the external URL.

+ -

| NAME |
|----------|
| EXSERVER |

Enter the domain name you will use with your external Client Access servers
(example:mail.contoso.com).





mail.exchange2016demo.com

The outcome of this approach is that all of the external URLs are configured to use that namespace, but the internal URLs remain untouched. This is not ideal for our goal of configuring all services to use the same internal and external namespace.

Instead you can edit the configuration of each virtual directory listed in the Exchange Admin Center by clicking the edit icon.

servers databases database availability groups **virtual directories** certificates

Select server: All servers
Select type: All

| NAME | SERVER | TYPE | VERSION | LAST MODIFIED T... |
|-------------------------------|----------|---------|---------------------------|--------------------|
| Autodiscover (Default W... | EXSERVER | Auto... | Version 15.1 (Build ... | 13/10/2015 9:2... |
| ecp (Default Web Site) | EXSERVER | ECP | Version 15.1 (Build 22... | 21/11/2015 1:40... |
| EWS (Default Web Site) | EXSERVER | EWS | Version 15.1 (Build 22... | 21/11/2015 1:40... |
| Microsoft-Server-ActiveSyn... | EXSERVER | EAS | Version 15.1 (Build 22... | 21/11/2015 1:40... |
| OAB (Default Web Site) | EXSERVER | OAB | Version 15.1 (Build 22... | 21/11/2015 1:40... |
| owa (Default Web Site) | EXSERVER | OWA | Version 15.1 (Build 22... | 21/11/2015 1:39... |
| PowerShell (Default Web Si... | EXSERVER | Powe... | Version 15.1 (Build 22... | 13/10/2015 9:26... |

Autodi
Authentic
Point Sec

From here you can edit both the internal and external namespaces for the virtual directory, as well as additional settings such as authentication.

owa (Default Web Site)

► **general**
authentication
features
file access

Server:
EXSERVER

Server version:
Version 15.1 (Build 225.42)

Website:
Default Web Site

Outlook Web App version:
Exchange2013

Last modified time:
21/11/2015 1:39 PM

Internal URL:
https://mail.exchange2016demo.com/owa

External URL:
https://mail.exchange2016demo.com/owa

This will achieve the desired outcome, but it is a slow and tedious task. For a single server it would be annoying, for multiple servers it would be downright frustrating. Also, if you ever needed to reconfigure the server you'd need to manually repeat the task.

Instead let's look at using PowerShell to make the namespace configuration changes.

Configuring Client Access Namespaces Using PowerShell

Each of the virtual directories on the Exchange 2016 server has a corresponding PowerShell cmdlet that is used to configure the virtual directory's settings, including the Autodiscover virtual directory even though we don't actually need to configure that one (instead we configure the AutodiscoverServiceInternalUri on the Client Access server).

For example, to configure the same URLs for OWA as shown in the screenshot above:

```
[PS] C:\>Get-OwaVirtualDirectory -Server EXSERVER | Set-OwaVirtualDirectory -InternalUrl https://mail.exchange2016demo.com/owa -ExternalUrl https://mail.exchange2016demo.com/owa
```

Because each virtual directory uses a different cmdlet this task can still become quite tedious as you move through each virtual directory on every server. But of course if the task can be performed in PowerShell it can be scripted!

Automating boring tasks is one of PowerShell's great strengths, and this task is no different. Since every virtual directory (and the Autodiscover service URI) can be configured in PowerShell we can write a script to perform the task quickly.

In your own environment you could manually write out each of the PowerShell commands for your server names and simply save them in a script file. Or you can use my [ConfigureExchangeURLs.ps1 script](#) with a few easy to use parameters.

Here's an example of how I can apply my desired namespace configuration to my Exchange 2016 server using ConfigureExchangeURLs.ps1.

```
C:\Scripts> .\ConfigureExchangeURLs.ps1 -Server EXSERVER -InternalURL mail.exchange2016demo.com -ExternalURL mail.exchange2016demo.com

-----
Configuring EXSERVER
-----

Values:
- Internal URL: mail.exchange2016demo.com
- External URL: mail.exchange2016demo.com
- Outlook Anywhere default authentication: NTLM
```


- Outlook Anywhere internal SSL required: True
- Outlook Anywhere external SSL required: True

Configuring Outlook Anywhere URLs

Configuring Outlook Web App URLs

WARNING: You've changed the InternalURL or ExternalURL for the OWA virtual directory. Please make the same change for

the ECP virtual directory in the same website.

Configuring Exchange Control Panel URLs

Configuring ActiveSync URLs

Configuring Exchange Web Services URLs

Configuring Offline Address Book URLs

Configuring MAPI/HTTP URLs

Configuring Autodiscover

Now let's look at the output of GetExchangeURLs.ps1 again.

```
PS C:\Scripts> .\GetExchangeURLs.ps1 -Server EXSERVER
```

```
-----  
Querying EXSERVER  
-----
```

Outlook Anywhere

- Internal: mail.exchange2016demo.com
- External: mail.exchange2016demo.com

Outlook Web App

- Internal: https://mail.exchange2016demo.com/owa
- External: https://mail.exchange2016demo.com/owa

Exchange Control Panel

- Internal: https://mail.exchange2016demo.com/ecp
- External: https://mail.exchange2016demo.com/ecp

Offline Address Book

- Internal: https://mail.exchange2016demo.com/OAB
- External: https://mail.exchange2016demo.com/OAB

Exchange Web Services

- Internal: https://mail.exchange2016demo.com/EWS/Exchange.asmx
- External: https://mail.exchange2016demo.com/EWS/Exchange.asmx

MAPI

- Internal: https://mail.exchange2016demo.com/mapi
- External: https://mail.exchange2016demo.com/mapi

ActiveSync

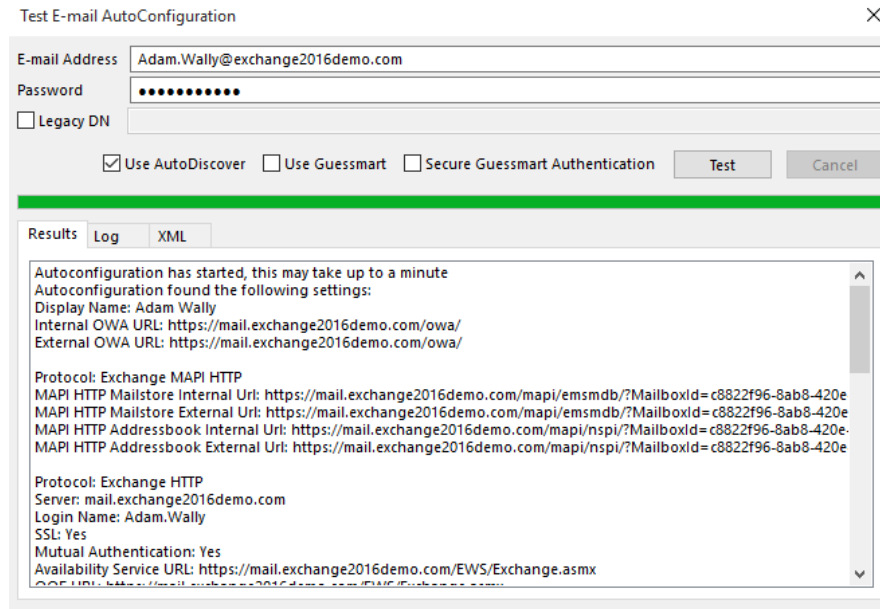
- Internal: https://mail.exchange2016demo.com/Microsoft-Server-ActiveSync
- External: https://mail.exchange2016demo.com/Microsoft-Server-ActiveSync

Autodiscover

- Internal SCP: https://mail.exchange2016demo.com/Autodiscover/Autodiscover.xml

Repeating the Outlook auto configuration test from earlier we can see that Autodiscover is now returning the new URLs to the clients as well. However, you may notice that:

- You still receive security warnings due to the SSL certificate currently in use
- The Autodiscover response doesn't change until an IISReset is performed on the Exchange server, and even then it may take several more minutes to take effect



As you can see there are several methods available for making the configuration changes, with the PowerShell script being the easiest by far. If it is your first time configuring a server it is worth doing the task manually the first time to gain some understanding of what is involved, but if you're planning to deploy multiple servers then using a script such as [ConfigureExchangeURLs.ps1](#) is highly recommended.

Now that the namespaces are configured the next step is to configure an SSL certificate for the server.

Exchange Server 2016 and SSL Certificates

Exchange Server 2016 communicates with clients, applications and other servers over a variety of network protocols such as HTTPS, SMTP, IMAP and POP. Much of this communication, particularly clients and applications, involves username and password-based authentication. When user credentials are sent over the network they are sent “in the clear”, meaning they can potentially be intercepted and read by an attacker. Other information transmitted during the session may also be sensitive and prone to abuse if interception was possible.

To secure these communications Exchange Server 2016 uses SSL certificates to encrypt the network traffic between the server, clients and applications. This includes:

- Outlook connecting to Outlook Anywhere (RPC-over-HTTP) or MAPI-over-HTTP
- Web browsers connecting to Outlook on the web (OWA)
- Mobile devices connecting to ActiveSync to access mailboxes and calendars
- Applications connecting to Exchange Web Services (EWS) for free/busy and other lookups
- Email clients connecting to secure POP or IMAP
- TLS encrypted SMTP between Exchange servers or other email servers

When Exchange Server 2016 is first installed it generates a self-signed SSL certificate that is then enabled for IIS (HTTPS services like OWA, EWS and ActiveSync), SMTP, POP and IMAP. The self-signed certificate allows the server to be “secure by default” and begin encrypting network communications right from the start, but it is only intended to be used temporarily while you provision the correct SSL certificates for your environment.

When deploying Exchange Server 2016 you should plan to replace the self-signed certificate with a valid SSL certificate for your deployment scenario. This involves an investment of anywhere from \$99 to several thousand dollars depending on your Client Access namespace scenario, the type of certificate you purchase, and which certificate authority you purchase it from.

If you’re tempted to stick with the self-signed certificate, or to try and disable SSL requirements on Exchange services, I strongly recommend you do not do those things.

- Deliberately trying to reduce the security of your Exchange environment is unwise
- The hours you’ll spend configuring and troubleshooting your attempted workarounds is costlier than just buying the correct SSL certificate
- Some stuff just flat out won’t work if you try to work around SSL requirements

SSL Certificate Requirements

There are three basic requirements for an SSL certificate in an Exchange Server 2016 deployment.

- **Correct server/domain names** – the SSL certificate must contain the namespaces (aka, URLs, aliases, domain names) to match the names that clients will be connecting to (for example, users typing “mail.exchange2016demo.com” in their web browser to access Outlook on the web)

- **Certificate validity period** – each SSL certificate has a fixed period of time during which it can be considered valid. When the SSL certificate reaches its expiry date it will need to be renewed to continue working.
- **Trusted certificate authority** – clients will only trust SSL certificates that have been issued by a certificate authority that they already trust. This is one reason that the self-signed certificate is not suitable for general production use, because your clients will not trust certificates issued by the Exchange server itself. There are a wide range of certificate authorities available to purchase certificates from that your client operating systems and devices will trust. I generally recommend using [DigiCert](#).

Choosing a certificate authority is simple enough, and the validity period will be determined by how long you purchase the certificate for (usually a minimum of 12 months, but the longer the validity period the less the certificate tends to cost per year). That leaves the server/domain names (or namespaces) as the main decision point when planning your SSL certificates.

Namespaces to Include on SSL Certificates

The simplest approach to namespaces for Exchange Server 2016 is to use a single namespace for all HTTPS services. An example of this approach was shown in the previous section on configuring namespaces.

In addition to the HTTPS namespace it is also common to use a separate namespace for each of the SMTP, POP and IMAP services, although it is certainly not required to do so. There is also the Autodiscover CNAME to consider, and the root domain as well.

In a simple environment where the domain name used for email addresses is **exchange2016demo.com**, and taking all of the above into consideration, the namespaces could be planned as:

- mail.exchange2016demo.com (for all HTTPS, SMTP, POP and IMAP)
- autodiscover.exchange2016demo.com (for the Autodiscover CNAME)
- exchange2016demo.com (for root domain Autodiscover lookups)

The recommended practice is to only include aliases as namespaces on SSL certificates, and not any server fully-qualified domain names (real server names). Due to recent changes to certificate issuance rules you may also find it impossible to request an SSL certificate for a

domain name that is not internet-routable or that you do not legitimately own (e.g., domain.local).

Which Type of Certificate to Purchase?

Certificate authorities such as [DigiCert](#) can sell you a variety of certificate types, and some certificate authorities have different names for what is basically the same thing.

A standard SSL certificate contains a single name and is generally the cheapest to purchase, however these are not suitable for even the simplest of namespace designs.

A wildcard SSL certificate allows you to secure multiple names on a domain without having to specify the exact names on the certificate itself. For example, a DigiCert wildcard certificate can be acquired for **exchange2016demo.com** and ***.exchange2016demo.com**. While these are often a lower cost option, wildcard certificates can have compatibility issues with some integration scenarios with other systems, as well as not being suitable for secure POP and IMAP configurations.

A SAN or UC (Unified Communications) certificate is the recommended type of SSL certificate to purchase. A SAN certificate can contain multiple names. For example, a DigiCert UC certificate can include up to four names at the normal price, with additional names up to 25 total being available at an additional cost. While the cost of a SAN/UC certificate will be more than a wildcard certificate you are less likely to run into any compatibility issues with the SAN/UC certificate, as long as the certificate contains the correct names. If you make an error with your namespace planning or need to add a name later DigiCert and some other providers will allow you to re-issue the certificate at no cost, while other providers will charge a re-issuing fee.

How Many SSL Certificates Should You Purchase?

After planning your namespaces and choosing a certificate authority you may be considering how many SSL certificates to purchase, especially if you have more than one Exchange 2016 server.

The recommend practice is to provision as few SSL certificates as possible, because it is simpler to administer as well as less costly to purchase. So while it is possible to have separate certificates for each of the HTTPS, SMTP, POP and IMAP services, it is recommended to use one certificate for all of them unless you have a specific scenario that requires different certificates.

Note that while only one SSL certificate can be enabled for HTTPS on a server, multiple SSL certificates can be enabled for SMTP. However, in most simple deployments only a single certificate will be required for SMTP.

Similarly, it is recommended to use the same SSL certificate for all of the Exchange servers that will be configured with the same namespaces. For example, if you have two Exchange 2016 servers in a site that will be load-balanced, and both have the “mail.exchange2016demo.com” namespace configured for HTTPS services, they should have the same certificate installed. You achieve this by provisioning the certificate on the first server, and then exporting it and importing it to as many additional servers as needed.

Configuring the SSL Certificate

There are two methods you can use for generating a certificate request for [Exchange Server 2016](#):

- The Exchange Admin Center (you can think of this as the GUI method)
- The Exchange Management Shell (or PowerShell, you can think of this as the command line method)

Generating the certificate request (or CSR) using the Exchange Admin Center is generally easier of the two options, and it’s the method that I’ll demonstrate now.

To begin, open your web browser and log in to the Exchange Admin Center. After logging in, navigate to **servers** and then **certificates**.

Enterprise Office 365 Administrator ?

Exchange admin center

recipients permissions compliance management organization protection mail flow mobile public folders unified messaging **servers**

servers databases database availability groups virtual directories **certificates**

Select server: EXSERVER.exchange2016demo.com

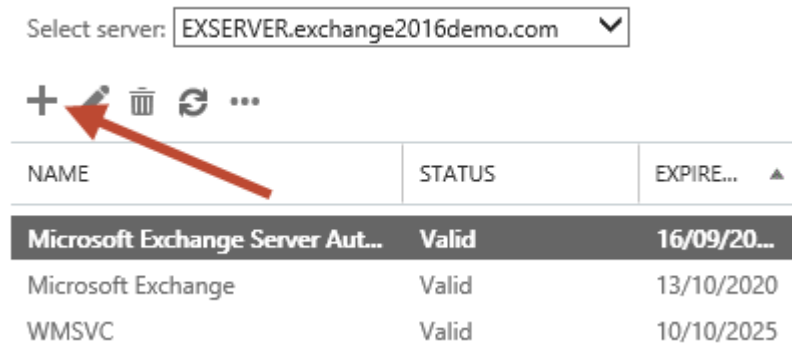
| NAME | STATUS | EXPIRE... |
|----------------------------------|--------|-------------|
| Microsoft Exchange Server Aut... | Valid | 16/09/20... |
| Microsoft Exchange | Valid | 13/10/2020 |
| WMSVC | Valid | 10/10/2025 |

Microsoft Exchange Server Auth Certificate

Self-signed certificate
Issuer: CN=Microsoft Exchange Server Auth (ertificate)

Status
Valid
Expires on: 16/09/2020
[Renew](#)

Click the “+” icon to start a new CSR.



Choose to create a request for a certificate from a certification authority.

new Exchange certificate

This wizard will create a new certificate or a certificate request file.

You can either create a self-signed certificate or request a certificate from a certification authority. [Learn more...](#)

- ☒ Create a request for a certificate from a certification authority
☐ Create a self-signed certificate

Enter a friendly name for the certificate. You’ll see this name in the list of certificates installed on the server, so make it something that you will easily recognise. For example, there’s already a self-signed certificate named “Microsoft Exchange”, so call your new certificate something different such as “Exchange 2016 SAN Certificate”.

new Exchange certificate

*Friendly name for this certificate:

Exchange 2016 SAN Certificate

Although wildcard certificates are generally supported for Exchange Server 2016 I am not going to be installing a wildcard certificate in this example.

new Exchange certificate

☐ Request a wildcard certificate. A wildcard certificate can be used to secure all sub-domains under your root domain with a single certificate. [Learn more](#)

*Root domain:

Choose a server to store the certificate request on. The same server is later used to complete the certificate request, and will be the first server that has the certificate installed. You can later export the certificate from this server and import it into other Exchange servers that have the same namespaces configured.

new Exchange certificate

*Store certificate request on this server:

| | | |
|----------|---|-----------|
| EXSERVER | X | Browse... |
|----------|---|-----------|

Next we select the domain names to include on the SSL certificate. You'll notice that the wizard has pre-populated the list based on the namespaces configured on the various Exchange services.

However, you may also notice if you scroll down that the server's real name is included in that list due to the default configuration of the POP and IMAP services, even if those services are not enabled. You can edit the entries at this step, but I find it easier to proceed to the next step and modify the list there instead.

new Exchange certificate

Specify the domains you want to be included in your certificate. [Learn more](#)



| ACCESS | DOMAIN | |
|---|-------------------------|---|
| Exchange ActiveSync (when accessed from the Internet) | mail.exchange2016de... | ^ |
| Exchange ActiveSync (when accessed from the intranet) | mail.exchange2016de... | |
| Autodiscover (when accessed from the Internet) | AutoDiscover.exchang... | |
| Autodiscover (when accessed from the intranet) | mail.exchange2016de... | |
| POP | EXSERVER | |
| IMAP | EXSERVER | |
| Outlook Anywhere | mail.exchange2016de... | v |

At the next step you can select and remove any unwanted names, edit existing names, or add more names to the certificate request. In this example I've modified the list to include only the planned namespaces:

- mail.exchange2016demo.com (for HTTPS services)
- autodiscover.exchange2016demo.com (the Autodiscover CNAME that may be used by non-domain joined devices such as mobile phones)
- exchange2016demo.com (the root domain, which is optional and depends on your specific scenario, but it's harmless to include it if you're not sure)

new Exchange certificate

Based on your selections, the following domains will be included in your certificate. You can add additional domains here, or make changes. [Learn more](#)



| DOMAIN |
|-----------------------------------|
| mail.exchange2016demo.com |
| AutoDiscover.exchange2016demo.com |
| exchange2016demo.com |

Enter your organization information for the certificate request. This information will form part of the validation process by the certificate authority that is issuing your certificate, so using correct and valid details is important. If any of the details are incorrect the certificate authority may contact you for additional proof of ownership before they'll issue you a certificate, slowing down the whole process.

new Exchange certificate

Specify information about your organization. This is required by the certification authority.

[Learn more](#)

*Organization name:

Exchange Server Pro

*Department name:

IT

*City/Locality:

Brisbane

*State/Province:

Queensland

*Country/Region name:

Australia

Enter a UNC path to save the certificate request to. The UNC path you provide must be accessible by the Exchange server's computer account, or by the Exchange Trusted Subsystem group. Simply choosing a UNC path that points to the Exchange server itself should be fine. You'll also need to be able to access the location yourself to be able to submit the request to the certificate authority.

new Exchange certificate

*Save the certificate request to the following file (example:

\\myservername\share\mycertrequest.REQ):

\\exserver\c\$\admin\exchangecertificate.req

You'll need to submit the contents of the file you entered to a certification authority.

After you receive the certificate file from the certification authority, you'll need to click Complete in the Information pane to install it on your Exchange server. [Learn more](#)

Click Finish, and the certificate request will be generated in the UNC path you chose.

You can now submit the CSR to a certificate authority such as [Digicert](#). When you've received your certificate, return to the Exchange Admin Center and complete the pending certificate request.

In the Exchange Admin Center navigate to **servers**, then **certificates**. Choose the Exchange server you need to complete the pending certificate request for, and selecting the pending request. Click the **Complete** link on the right side of the page.

Select server: EXSERVER.exchange2016demo.com

+ ✎ 🗑️ ↺ ...

| NAME | STATUS | EXPIRE... | |
|------------------------------------|-----------------|-------------|---|
| Exchange 2016 SAN Certificate | Pending request | 14/10/20... | <div>Exchange 2016 SAN Certificate</div> <div>Certification authority-signed certificate Issuer: C=AU, S=Queensland, L=Brisbane, O=LockLAN Systems Pty Ltd, OU=IT, CN=mail.exchange2016demo.com</div> <div>Status</div> <div>Pending request Expires on: 14/10/2016</div> <div>Complete</div> |
| Microsoft Exchange Server Auth ... | Valid | 16/09/2020 | |
| Microsoft Exchange | Valid | 13/10/2020 | |
| WMSVC | Valid | 10/10/2025 | |

Enter the UNC path to where you saved the certificate file provided by the CA, and click **OK**.

complete pending request

This will import the certificate file that you received from the certification authority. After it's imported, you can assign this certificate to various Exchange services. [Learn more](#)

*File to import from (example: \\server\folder\MyCertificate.CER):

\\exserver\c\$\admin\certnew.cer

If the import is successful, the certificate should now appear with a status of “Valid”.

Select server: EXSERVER.exchange2016demo.com

+ ✎ 🗑️ ↺ ...

| NAME | STATUS | EXPIRE... |
|------------------------------------|--------|-------------|
| Exchange 2016 SAN Certificate | Valid | 14/10/20... |
| Microsoft Exchange Server Auth ... | Valid | 16/09/2020 |
| Microsoft Exchange | Valid | 13/10/2020 |
| WMSVC | Valid | 10/10/2025 |

When an [SSL certificate](#) has been installed for Exchange Server 2016 you need to assign it to Exchange services before it will be used.

Select server: EXSERVER.exchange2016demo.com ▼

+ ✎ 🗑️ ↺ ...

| NAME | STATUS | EXPIRE... ▲ | |
|--------------------------------------|--------------|--------------------|--|
| Exchange 2016 SAN Certificate | Valid | 14/10/20... | <p>Exchange 2016 SAN Certificate</p> <p>Certification authority-signed certificate Issuer: CN=exchange2016demo-DC-CA, DC=exchange2016demo, DC=com</p> <p>Status</p> <p>Valid Expires on: 14/10/2017 Renew</p> <p>Assigned to services NONE</p> |
| Microsoft Exchange Server Auth ... | Valid | 16/09/2020 | |
| Microsoft Exchange | Valid | 13/10/2020 | |
| WMSVC | Valid | 10/10/2025 | |

Select the SSL certificate and click the edit icon.

Select server: EXSERVER.exchange2016demo.com ▼

+ ✎ 🗑️ ↺ ...

| NAME | STATUS | EXPIRE... ▲ |
|--------------------------------------|--------------|--------------------|
| Exchange 2016 SAN Certificate | Valid | 14/10/20... |

Select services, then tick the boxes for each service you wish to enable.

- IIS is used for all HTTPS services (such as OWA, ActiveSync, Outlook Anywhere). Only one certificate can be assigned to IIS, so it's important that the certificate contains all of the correct names configured as URLs for your HTTPS services.
- SMTP is used for TLS-encrypted mail flow. More than one certificate can be assigned to SMTP.
- POP and IMAP are disabled by default in Exchange Server 2016, but if you are planning to enable them you should assign a certificate, whether that is the same certificate used for HTTPS or a different one.

- UM is optional as well. If you are planning to use the UM features of Exchange Server 2016 enable a certificate for UM as well, again that can be the same certificate as used for HTTPS services or a different one.

Exchange 2016 SAN Certificate

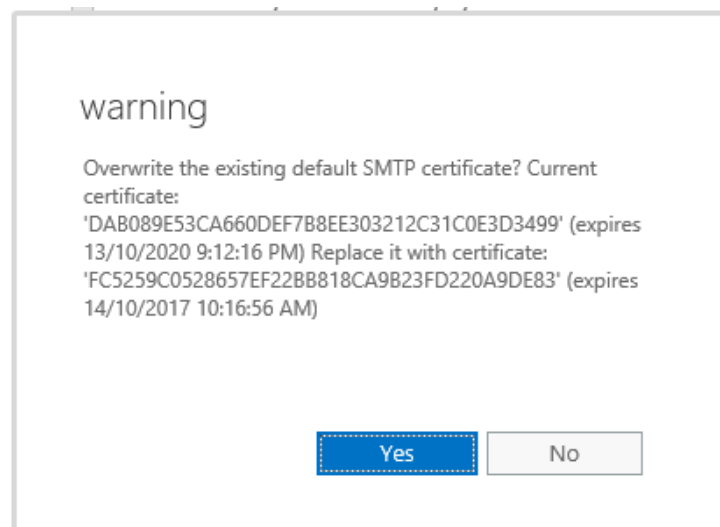
general

► **services**

Specify the services you want to assign this certificate to. [Learn more](#)

- ☒ SMTP
- ☐ Microsoft Exchange Unified Messaging
- ☐ Unified Messaging Call Router
- ☐ IMAP
- ☐ POP
- ☒ IIS

Click **Save** when you've select the services you need to use the SSL certificate for. If you are assigning an SMTP certificate you may be prompted to overwrite the default SMTP certificate. SMTP can have multiple certificates assigned, and for a simple deployment where the single SSL certificate you acquired contains the SMTP namespace you plan to use on connectors it is generally fine to say **Yes** to this prompt.



After you've completed those steps the SSL certificate will be used by Exchange for those services you selected.

Testing the Client Access Configuration

If everything has been configured correctly, you should be able to see the following behaviour:

- Outlook opens and connects to Exchange mailboxes with no security warnings or unexpected authentication prompts.
- You can use a web browser to connect to the OWA namespace (<https://mail.exchange2016demo.com/owa> in this example) and log in to Outlook on the web without any browser errors or certificate warnings.
- From within Outlook you can access and modify the Out of Office settings (this relies on successful connectivity to Exchange Web Services, so it is a good test).

If you don't have immediate success, then you should:

- Double-check your configuration to make sure you've actually configured the namespaces and SSL certificate correctly.
- Perform an IISReset on the server, or restart the client computer you're testing from
- Delete and recreate the Outlook profile to test a newly created profile.

Sometimes in the real world there is simply a delay for the configuration changes to take effect. This is one reason why it is better to perform these changes immediately when the server has been deployed, and before any users are set up to use it.

What About External Access?

We've looked at configuring Client Access and testing it for an internal client, but what about external access? Services such as Outlook on the web and ActiveSync are more likely to be used by people outside of the corporate network, so we need to make sure they're accessible from there as well.

For external access to Client Access services to work there are three main components:

- Public DNS records for the Client Access namespaces. We've already added one for the mail.exchange2016demo.com namespace, but we will also need to add the Autodiscover.exchange2016demo.com record to the public DNS zone as well so that non-domain members (such as mobile devices) can still locate the Autodiscover server.


- A firewall rule that allows TCP port 443 (for HTTPS traffic) and NATs it to the Exchange server.
- A trusted SSL certificate. I'm mentioning this one again because it is possible to use an internal certificate authority to issue the Exchange certificate, which will work for domain-joined computers that trust the enterprise CA. But this will not work for non-domain members such as mobile devices and home computers, which is why it is always recommended to use a public CA to acquire the SSL certificate.

Proceed with the addition of the Autodiscover DNS record and the configuration of your firewall by following the guidance that your DNS host and firewall vendor provided you.


To test external access, if you do not have an external computer or mobile device to test with, you can use the [Microsoft Remote Connectivity Analyzer](#).

Select the test you want to run.


Exchange Server
Lync / OCS Server
Office 365
Client
Message Analyzer


Microsoft Exchange ActiveSync Connectivity Tests


- ☒ Exchange ActiveSync
- ☐ Exchange ActiveSync Autodiscover


Microsoft Exchange Web Services Connectivity Tests

- ☐ Synchronization, Notification, Availability, and Automatic Replies
- ☐ Service Account Access (Developers)


Microsoft Office Outlook Connectivity Tests

- ☐ Outlook Connectivity
- ☐ Outlook Autodiscover


Internet Email Tests

- ☐ Inbound SMTP Email
- ☐ Outbound SMTP Email
- ☐ POP Email
- ☐ IMAP Email

The Exchange ActiveSync and Outlook Connectivity tests will both test your Client Access configuration. The output of the tests is very detailed and will help you to zoom in on the potential causes of a failed test.

Configuring Transport

Transport for Exchange Server 2016 refers to mail flow, or the passage of email messages between different recipients inside and outside of your organization.

A newly installed Exchange 2016 server, with no other Exchange servers in the environment, will only work for internal mail flow, for example from one mailbox on the server to another mailbox on the server. A newly installed Exchange 2016 server in an environment with other existing Exchange servers will normally be able to begin participating in all mail flow without any configuration at all. However, if you plan to decommission the other servers then the information in this chapter about configuring transport will still be very important for you to understand.

Let's take a look at the three most common transport requirements:

- Inbound mail flow from the internet to recipients on the Exchange server
- Outbound mail flow to the internet from senders on the Exchange server
- Applications and systems using the Exchange server for SMTP relay

Inbound Mail Flow

Configuring inbound mail flow for an [Exchange Server 2016](#) environment is reasonably simple, however there are several different parts involved. For your server to receive email from the internet and deliver it to internal recipients there needs to be:

- An Accepted Domain configured for the organization
- An email address assigned to the recipient
- MX records in your public DNS zone
- SMTP connectivity from external senders to your Exchange server, or a mail route that leads to your Exchange server

The Exchange server will accept SMTP connections using a receive connector. A receive connector that is suitable for incoming email from the internet is pre-configured for you by Exchange setup, so there's no need for you to configure one yourself. The receive connector is named **Default Frontend SERVERNAME**.

rules delivery reports accepted domains email address policies **receive connectors**
send connectors

Select server: EXSERVER.exchange2016demo.com ▼

+ ✎ 🗑️ ↺ ⋮

| NAME ▲ | STATUS | ROLE | |
|----------------------------------|----------------|--------------------------|---|
| Client Frontend EXSERVER | Enabled | FrontendTransport | Default Frontend EXSERVER Last modified: 13/10/2015 9:25:43 PM Version: Version 15.1 (Build 225.42) Connector status - Enabled Disable Logging - On Off Maximum receive message size (MB): 36 |
| Client Proxy EXSERVER | Enabled | HubTransport | |
| Default EXSERVER | Enabled | HubTransport | |
| Default Frontend EXSERVER | Enabled | FrontendTransport | |
| Outbound Proxy Frontend EXS... | Enabled | FrontendTransport | |

If you look at the properties of that connector you might notice that “Anonymous Users” is enabled as a permission group. Yes, this is the correct configuration for the connector, and no that does not mean it can be abused as an open relay.

Configuring Accepted Domains

Accepted domains define which domain names your Exchange servers will accept email for. When you [install a new Exchange 2016 server](#) the DNS name of the Active Directory forest is automatically added as an accepted domain for the Exchange organization. If your Active Directory forest DNS name happens to match the SMTP domain you plan to use for email, then there’s no additional work required here. Similarly, if you’re installing Exchange 2016 into an existing Exchange organization, the accepted domains are likely already configured.

You can view your accepted domains in the Exchange Admin Center. Navigate to **mail flow** and then choose **accepted domains**. In my example environment the accepted domain of exchange2016demo.com is already present, because it is also the namespace for the Active Directory forest, so it was added automatically by Exchange setup.

Exchange admin center

recipients permissions compliance management organization protection **mail flow**

rules delivery reports **accepted domains** email address policies send connectors

+ ✎ 🗑️ 🔍 ↻

| NAME | ACCEPTED DOMAIN | DOMAIN TYPE |
|--------------------------|----------------------|---------------|
| exchange2016demo.com ... | exchange2016demo.com | Authoritative |

If you need to add a new accepted domain click the “+” icon, which launches a wizard for the task. Enter a name for the accepted domain, then the domain name itself (I always just configure those two values to be the same).

new accepted domain

Accepted domains are used to define which domains will be accepted for inbound email routing.

*Name:

newdomain.com

*Accepted domain:

newdomain.com

This accepted domain is:

- ☒ **Authoritative:** Email is delivered only to valid recipients in this Exchange organization. All email for unknown recipients is rejected.
- ☐ **Internal Relay:** Email is delivered to recipients in this Exchange organization or relayed to an email server at another physical or logical location.
- ☐ **External Relay:** Email is relayed to an email server at another physical or logical location.

Notice the three options for the type of domain. The explanations are very clear, but to summarise:

- **Authoritative** – a domain for which your servers host the only recipients. For most scenarios this will be the correct choice.
- **Internal relay** – a domain for which your servers host some, but not all of the recipients. A typical use case for this type of accepted domain is a shared SMTP namespace, which is often required when two companies are merging or separating.

- External relay – a domain for which your server receives email, but hosts none of the recipients.

Add any domain names that you need for your organization, then move on to the email address policies.

Configuring Email Address Policies

The next step is to add email addresses to recipients in your organization. You can do this on a per-recipient basis, by simply opening the properties of the recipient (such as a mailbox), selecting **email address**, and adding the desired SMTP address.

Adam Wally

- general
- mailbox usage
- contact information
- organization
- ▶ **email address**
- mailbox features
- member of
- MailTip
- mailbox delegation

Each email address type has one default reply address. The default reply address is displayed in bold. To change the default reply address, select the email address that you want to set as the default, and then double-click to edit it.

Email address:

+ ✎ -

| TYPE | EMAIL ADDRESS |
|------|--------------------------------------|
| SMTP | adam.wally@exchange2016dem... |

☒ Automatically update email addresses based on the email address policy applied to this recipient

Of course this is not a very efficient way to manage multiple recipients, and even though PowerShell is available for automating this step, the more effective method is to use email address policies. An email address policy is configured by default when you install a new Exchange 2016 server, or it will simply use the existing policy if you're installing into an existing organization. Email address policies are found in the **mail flow** section of the Exchange Admin Center.

Exchange admin center

The screenshot shows the Exchange Admin Center interface. On the left is a navigation pane with links to recipients, permissions, compliance management, organization, protection, mail flow (highlighted), and mobile. The main area shows the 'email address policies' page. At the top, there are links for rules, delivery reports, accepted domains, email address policies (highlighted), and receive connectors. Below these are icons for adding, editing, deleting, and moving policies. A table lists the policies:

| NAME | PRIORITY | STATUS |
|----------------|----------|---------|
| Default Policy | Lowest | Applied |

To the right of the table, details for the 'Default Policy' are shown: Email Address Format, SMTP, and Primary: @exchange2016demo.com.

In my example environment the default email address policy configured by Exchange setup already contains the default accepted domain that was also configured by setup. The default address format is **alias@domain**, and we can either change that or add more address formats or addresses for different domain names to the policy if required.

The screenshot shows a web browser window titled 'Email Address Policy - Internet Explorer' with the URL <https://mail.exchange2016demo.com/ecp/EmailAddressPolicy/EditEmail>. The page displays the 'Default Policy' configuration. On the left, a sidebar shows 'general', 'email address format' (selected), and 'apply to'. The main content area shows the 'Email address format' configuration. It includes a section for '*Email address format:' with a table for adding new formats:

| TYPE | ADDRESS FORMAT |
|------|----------------|
| SMTP | @exchange2016 |

Below the table, there is a section for 'email address format' with a dropdown menu to 'Select an accepted domain:' (currently showing 'exchange2016demo.com'). There is also an option to 'Specify a custom domain name for the email address:'. Below this, there is a list of email address formats with radio buttons to select one. The first option, 'alias@contoso.com', is selected. Other options include 'John.Smith@contoso.com', 'JSmith@contoso.com', 'JohnS@contoso.com', 'Smith.John@contoso.com', 'SJohn@contoso.com', and 'SmithJ@contoso.com'. There is a link for 'More options...' and a checkbox for 'Make this format the reply email address'.

Earlier you may have noticed the check box on the mailbox user that says:

“Automatically update email addresses based on the email address policy applied to this recipient.”

In effect this means that the email address policy shown above will stamp the SMTP addresses on that recipient (and all the other recipients with that check box enabled), without me having to add them manually.

Review or modify your email address policies and confirm that recipients have the desired SMTP addresses, then move on to DNS records.

Configuring MX Records in DNS

With the accepted domains and email addresses configured the next thing to look at is the MX records in the public DNS zone.

MX stands for “mail exchanger”. An MX record is a type of DNS record, so any understanding of MX records has to begin with an understanding of the fundamentals of the Domain Name System (DNS).

The most important role of DNS for the majority of us is translating names into IP addresses so that network communications can occur.

For example, when you type www.microsoft.com into your web browser, DNS is used to look up that name to determine the IP address of the server to connect to. The domain name in this example is microsoft.com.

So if that is how a simple web browser connection is made, what about when somebody sends email to an @microsoft.com address?

Again DNS comes into play, but this time the look up is slightly different. The sending mail server will look up the MX record in DNS by following a sequence along these lines:

1. Look up the authoritative name servers for microsoft.com
2. Query the microsoft.com name servers for the MX records
3. Look up the names of the MX records in DNS to get their IP addresses

There can be multiple MX records for a domain, each with a different preference value. The preference is basically a way of setting the priority of each MX record. The lowest preference is the MX with the highest priority, i.e. the one that a sending mail server should try first.

The purpose of multiple MX records is to either:

- Provide some load balancing by using multiple MX records with the same preference set
- Provide a backup MX that can be used if the primary one is unavailable

The backup MX may be another mail server in your organization at a secondary site that has less bandwidth available to it. Or it could be a server hosted by a third party that provides

backup MX services. Either way the purpose is to give sending email systems somewhere to send messages rather than have to store them and retry later.

At least one MX record is required for other email systems to be able to locate yours in DNS. The steps to add the MX record to your DNS zone will vary depending on the DNS control panel your provider gives you access to. Basically you will need to configure:

- An MX record that resolves to an A record, for example mail.exchange2013demo.com
- The A record that resolves to an IP address on your firewall, or to a cloud-hosted email security service if you are choosing to use one for filtering spam and malware from your emails.

| | | | |
|--------------------------|----|---|-------------------------------|
| <input type="checkbox"/> | MX | exchange2016demo.com. | 40 mail.exchange2016demo.com. |
| <input type="checkbox"/> | A | mail.exchange2016demo.com. | 203.206.161.219 |

You can use the tools at MXToolbox.com to test your MX records.

The screenshot shows the MXToolbox website interface. At the top is the logo "MX TOOLBOX®". Below it is a navigation bar with links: Home, MX Lookup, Blacklists, Diagnostics, Domain Health, and Analyze Headers. The main section is titled "SuperTool Beta7". There is a search bar containing "exchange2016demo.com" and a button labeled "MX Lookup". Below this, the results for "mx:exchange2016demo.com" are displayed, including a "Find Problems" button. A table shows the MX record details:

| Pref | Hostname | IP Address | TTL | |
|------|--|--|--------|---------------------------------|
| 40 | mail.exchange2016demo.com | 203.206.161.219 | 60 min | Blacklist Check |

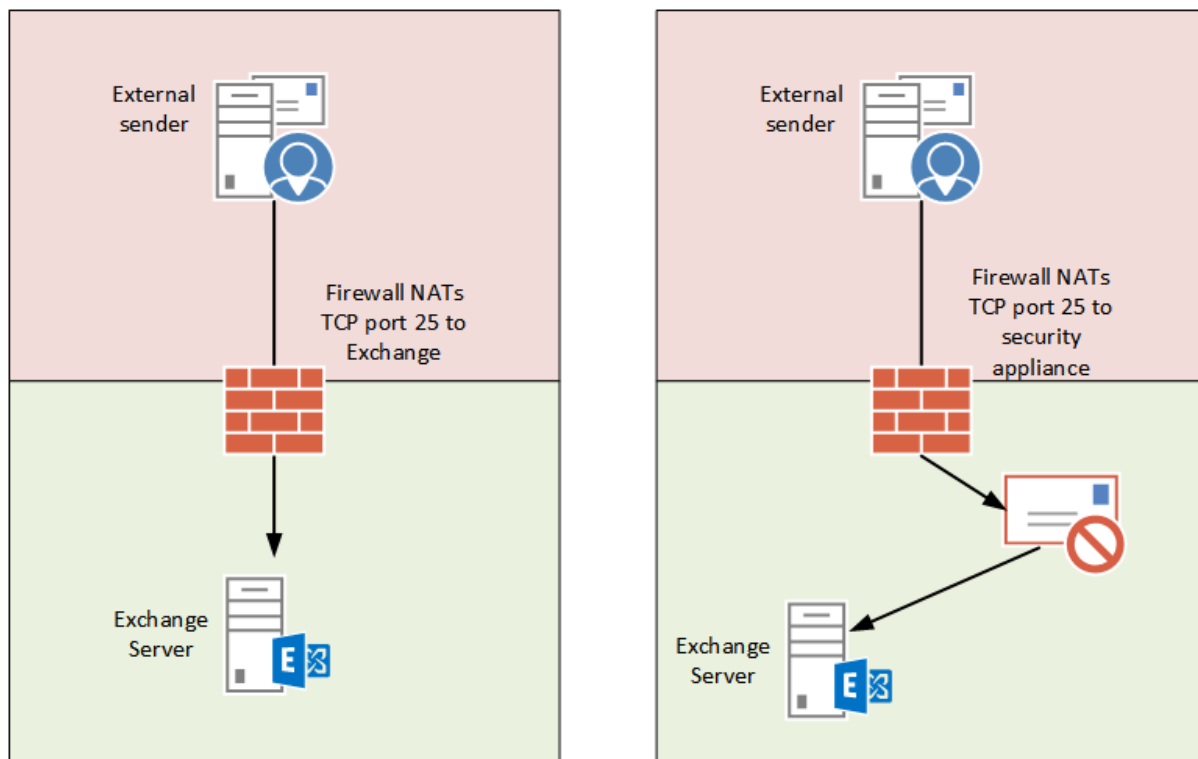
Below the table are links for "dns lookup", "dns check", "whois lookup", "spf lookup", and "dns prop". At the bottom, it says "Reported by ns1.uber.com.au on 10/19/2015 at 12:40:37 PM (UTC 0), [just for you](#). ([History](#))".

Configure and test your DNS records, then move on to SMTP connectivity.

Configuring SMTP Connectivity to the Exchange Server

The final piece of the solution is to establish SMTP connectivity to the Exchange server. There's generally two approaches used for this:

- The firewall is configured to NAT and allow SMTP connections directly to the Exchange server (either the Mailbox server or an Edge Transport server)
- SMTP connections first go to an inbound smart host, such as an email security appliance or cloud service, which then routes the messages on to your Exchange server



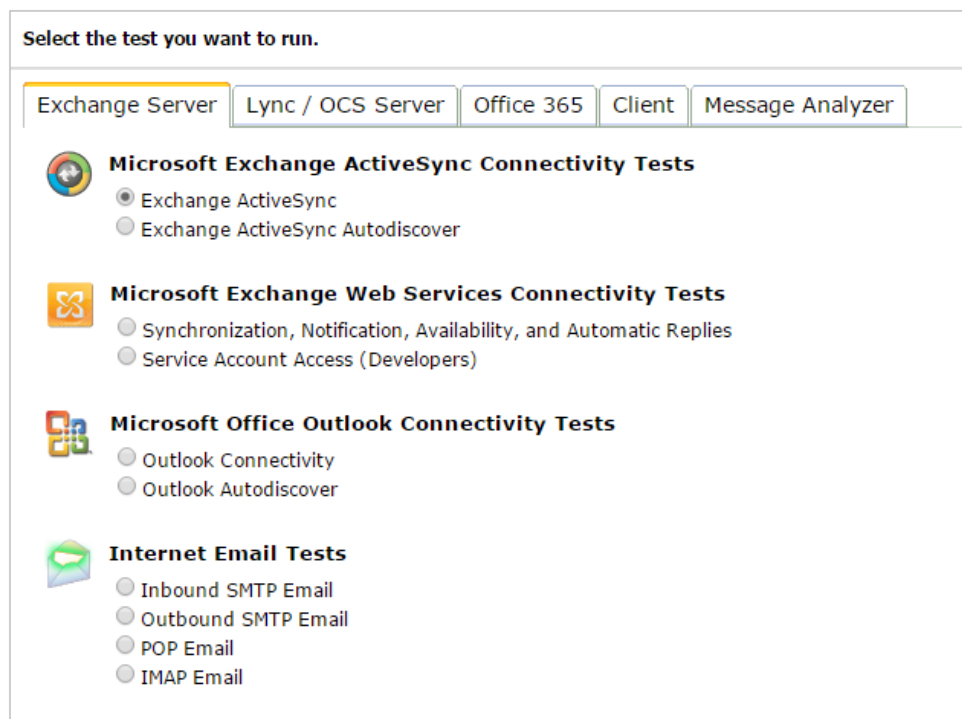
Of course, there are many other variations of how inbound SMTP connectivity is established depending on the size and complexity of the organization, but those are two typical examples.

The configuration steps for your firewall will depend on the type of firewall you're running. After configuring your firewall, you can look at performing tests of your end to end solution.

Testing Inbound Mail Flow


The simplest way to test inbound mail flow is of course to send an email from an external sender (such as a Gmail account) to a recipient on your Exchange server. If the email arrives, then inbound mail flow works!

If the email doesn't arrive, then a more useful test can be performed using the [Microsoft Remote Connectivity Analyzer](#).




Select the test you want to run.


Exchange Server | Lync / OCS Server | Office 365 | Client | Message Analyzer

 **Microsoft Exchange ActiveSync Connectivity Tests**


- ☒ Exchange ActiveSync
- ☐ Exchange ActiveSync Autodiscover

 **Microsoft Exchange Web Services Connectivity Tests**

- ☐ Synchronization, Notification, Availability, and Automatic Replies
- ☐ Service Account Access (Developers)

 **Microsoft Office Outlook Connectivity Tests**

- ☐ Outlook Connectivity
- ☐ Outlook Autodiscover

 **Internet Email Tests**

- ☐ Inbound SMTP Email
- ☐ Outbound SMTP Email
- ☐ POP Email
- ☐ IMAP Email

Use the Inbound SMTP Email test, which will provide much more detail about any of the parts of your configuration that are not working correctly.

One thing to keep in mind is that if you've *modified* your MX records as part of this configuration, then it's possible that other email servers on the internet won't start using your new MX record until the old one has expired from their DNS cache. This depends on the "time to live" (TTL) value of your MX records, which is usually 24 hours by default. You can lower this value to something more like 5 minutes, but that won't speed things up very much for those that already have the 24 hour TTL in their cache.

Fortunately, the Remote Connectivity Analyzer does not cache DNS records, so it should show you the current configuration. You may just need to wait longer for real world email servers to pick up the changes.

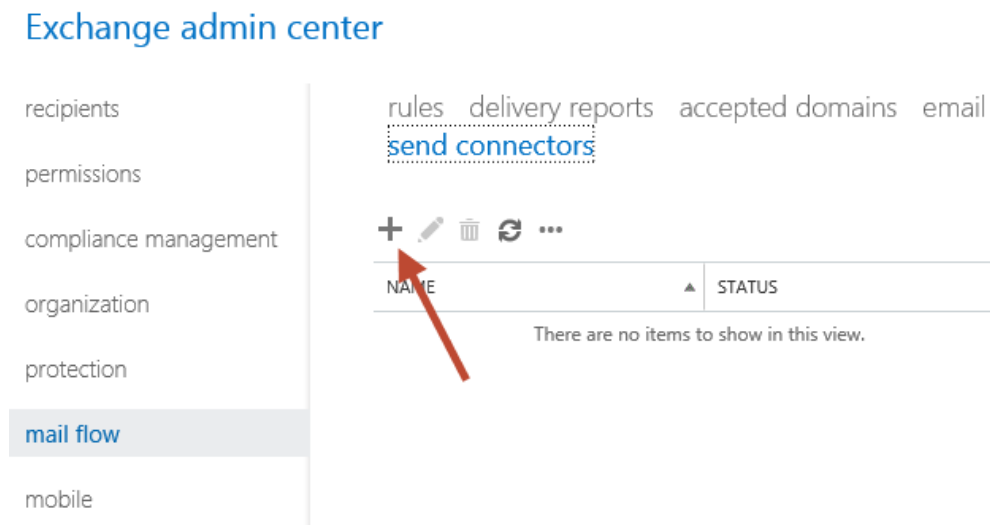
Outbound Mail Flow

When you first install Exchange Server 2016 there is no outbound mail flow configured by setup. If you happen to be installing into an existing Exchange organization then the existing outbound routes for the organization will apply, and mail sent by mailboxes on your new Exchange server to external recipients will likely work. However, if you're installing into a new organization, or want to change your existing outbound mail flow, then you'll need to create a send connector.

Send connectors control outgoing mail flow from your Exchange server. Every organization that needs to send email message to external recipients will need at least one send connector.

Creating a Send Connector

Log on to your Exchange Admin Center and navigate to **mail flow** and then **send connectors**. Click the "+" icon to create a new connector.



Give the new send connector a meaningful name and set the **Type** to **Internet**.

new send connector

Create a Send connector.

There are four types of send connectors. Each connector has different permissions and network settings. [Learn more...](#)

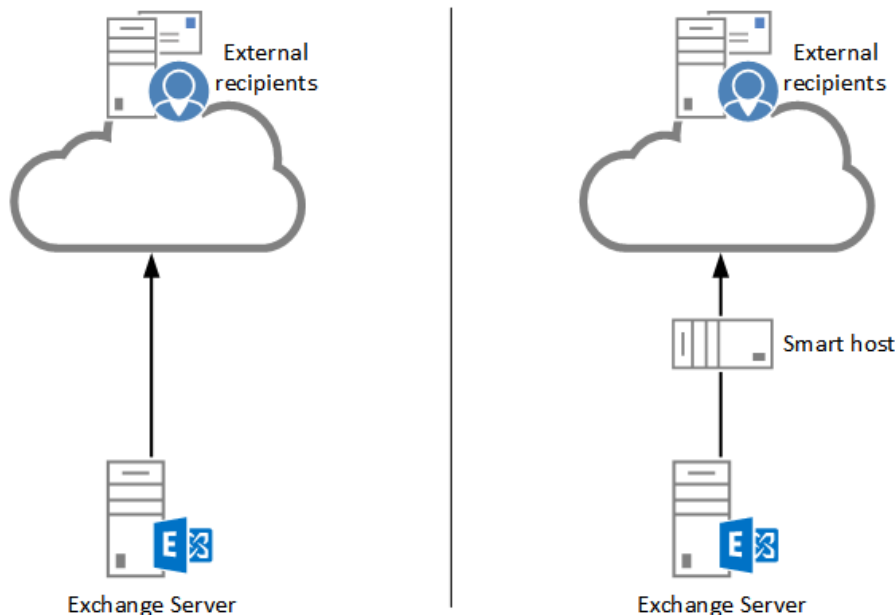
*Name:

Internet Email

Type:

- ☐ Custom (For example, to send mail to other non-Exchange servers)
- ☐ Internal (For example, to send intranet mail)
- ☒ Internet (For example, to send internet mail)
- ☐ Partner (For example, to route mail to trusted third-party servers)

Next you'll need to decide how the outbound emails will be delivered. There are two choices – by MX record, or via smart host. MX record delivery involves your Exchange server looking up the MX records of the recipient's domain in DNS, and then connecting directly to their email server via SMTP to deliver the email message. Smart host delivery involves your Exchange server sending the messages to a specified IP address or host name for another system (typically an email security appliance or cloud service) that is then responsible for the further delivery of that email message.



For this example, I'm going to use MX records to deliver the message. My server already has outbound firewall access on TCP port 25, and can resolve MX records on the internet using

DNS, so at a basic level this should work fine. There are other considerations such as SPF and IP reputation in the real world that may impact the delivery of email messages from your server.

new send connector

A send connector can route mail directly through DNS or redirect it to a smart host. [Learn more...](#)

*Network settings:

Specify how to send mail with this connector.

- ☒ MX record associated with recipient domain
☐ Route mail through smart hosts

+ ✎ -

| SMART HOST |
|------------|
| |

☐ Use the external DNS lookup settings on servers with transport roles

Set the address space for the send connector. An address space of “*” means “any domain” and is suitable if you have one send connector that is used for all outbound mail flow. You can use this address space option if you later need to configure specific send connectors for different domains.

new send connector

A Send connector routes mail to a specified list of domains. These domains can be an SMTP address space or a custom type. [Learn more...](#)

*Address space:

Specify the address space or spaces to which this connector will route mail.

+ ✎ -

| TYPE | DOMAIN | COST |
|------|--------|------|
| SMTP | * | 1 |
| | | |

☐ Scoped send connector

Finally, set the source server for the send connector. If you have multiple servers that you want to be responsible for outbound mail flow you can add more than one server to this list.

new send connector

A send connector sends mail from a list of servers with transport roles or Edge Subscriptions.
[Learn more...](#)

*Source server:
Associate this connector with the following servers containing transport roles. You can also add Edge Subscriptions to this list.

| SERVER | SITE | ROLE |
|----------|--|-----------------|
| EXSERVER | exchange2016demo.com/Configuration/Sites/... | Mailbox, Cli... |
| | | |

Click **Finish** to complete the wizard.

Testing the Send Connector

A simple test to verify that the send connector is working is to send an email from a mailbox on the server to an external address. If the email message is received by the external mailbox you can then check the message headers by copying them from the message and pasting them into the Message Analyzer at [ExRCA.com](#). This will verify for you that the email message took the intended route (via your new server) instead of some other existing outbound route in your organization.

| Received headers | | | | | |
|------------------|--|---|-----------------------|------------|-----------------------------|
| Hop: | Submitting host | Receiving host | Time | Delay | Type |
| 1 | EXSERVER.exchange2016demo.com ([fe80::4982fcb9:cd8b:5a27]) | EXSERVER.exchange2016demo.com ([fe80::4982fcb9:cd8b:5a27%12]) | 19/10/2015 3:02:07 pm | | mapi |
| 2 | EXSERVER.exchange2016demo.com (192.168.0.31) | EXSERVER.exchange2016demo.com (192.168.0.31) | 19/10/2015 3:02:07 pm | 0 seconds | Microsoft SMTP Server (TLS) |
| 3 | EXSERVER.exchange2016demo.com (203-206-161-219.perm.iinet.net.au. [203.206.161.219]) | mx.google.com | 19/10/2015 3:02:22 pm | 15 seconds | ESMTPS |
| 4 | | 10.25.207.15 | 19/10/2015 3:02:25 pm | 3 seconds | SMTP |

If the email message was not received check the transport queue on the Exchange 2016 server.

```
[PS] C:\>Get-Queue
```

| Identity | DeliveryType | Status | MessageCount | OutboundIPPool | NextHopDomain |
|---------------------|--------------|--------|--------------|----------------|---------------|
| ----- | ----- | ----- | ----- | ----- | ----- |
| EXSERVER\3 | DnsConnec... | Ready | 0 | 0 | gmail.com |
| EXSERVER\Submission | Undefined | Ready | 0 | 0 | Submission |

If you see message stuck in the queue for the next hop domain that you're trying to send to you can see more details about them by piping the command to Get-Message.

```
[PS] C:\>Get-Queue | Get-Message | fl
```

In particular look for the LastError attribute of the queued messages, which will often contain a status code that will tell you why the messages are not being delivered.

Since outbound mail flow depends on DNS and firewall access you can also check those items. For example, to verify that MX records can be resolved in DNS by the Exchange server use the Resolve-DnsName cmdlet.

```
[PS] C:\>Resolve-DnsName gmail.com -Type MX
```

You can also test SMTP connectivity from the server using Telnet. Because the Telnet client is not installed by default on Windows Server you may need to install it first.

```
[PS] C:\>Install-WindowsFeature Telnet-Client

Success Restart Needed Exit Code      Feature Result
-----
True      No                Success      {Telnet Client}
```

From a CMD prompt try to telnet to one of the MX records you resolved earlier.

```
C:\>telnet gmail-smtp-in.l.google.com 25

220 mx.google.com ESMTP bv3si49894863pbd.105 - gsmt
```

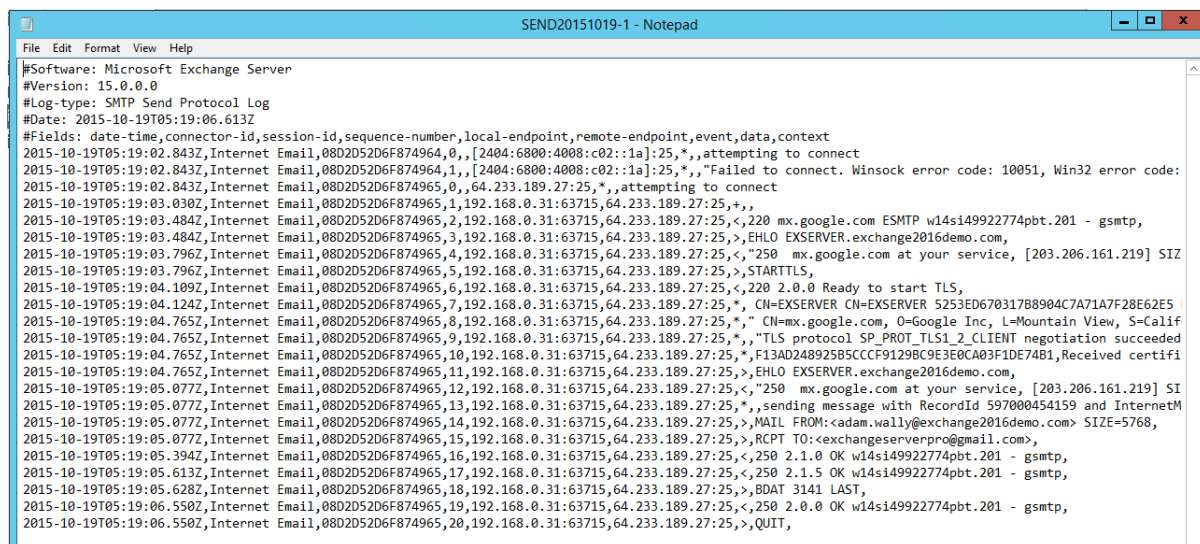
If you do not see the 220 response and banner, you may have an outbound SMTP connectivity issue that you need to look into further on your firewall.

Finally, if SMTP connectivity looks fine but the emails are still not being delivered you can enable protocol logging on your send connector and then use the log data to assist your troubleshooting.

```
[PS] C:\>Set-SendConnector "Internet Email" -ProtocolLoggingLevel Verbose
```

The protocol logs are stored by default in **C:\Program Files\Microsoft\Exchange Server\V15\TransportRoles\Logs\Hub\ProtocolLog\SmtpSend** and can be opened and read in

a text editor such as Notepad. The protocol log will show the SMTP conversation between your server and the external recipient's server, so any SMTP errors should appear in the log.



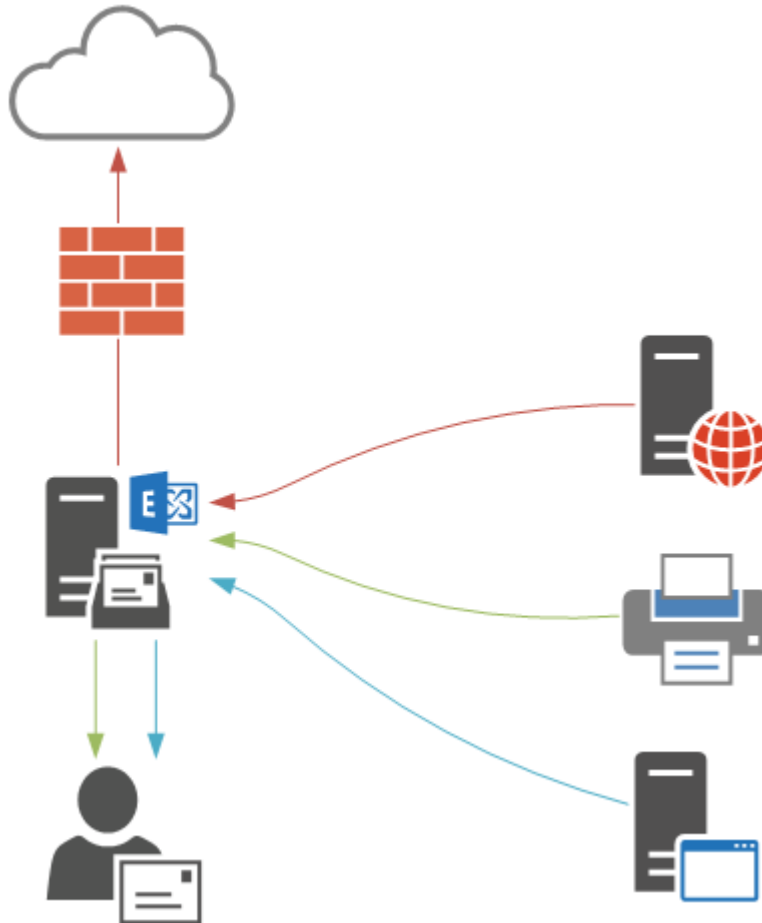
```
#Software: Microsoft Exchange Server
#Version: 15.0.0.0
#Log-type: SMTP Send Protocol Log
#Date: 2015-10-19T05:19:06.613Z
#Fields: date-time,connector-id,session-id,sequence-number,local-endpoint,remote-endpoint,event,data,context
2015-10-19T05:19:02.843Z,Internet Email,0802D52D6F874964,0,,[2404:6800:4008:c02::1a]:25,*,,attempting to connect
2015-10-19T05:19:02.843Z,Internet Email,0802D52D6F874964,1,,[2404:6800:4008:c02::1a]:25,*,,Failed to connect. Winsock error code: 10051, Win32 error code:
2015-10-19T05:19:02.843Z,Internet Email,0802D52D6F874965,0,,64.233.189.27:25,*,,attempting to connect
2015-10-19T05:19:03.030Z,Internet Email,0802D52D6F874965,1,192.168.0.31:63715,64.233.189.27:25,*,,
2015-10-19T05:19:03.484Z,Internet Email,0802D52D6F874965,2,192.168.0.31:63715,64.233.189.27:25,<,220 mx.google.com ESMTP w14si49922774pbt.201 - gsmtpt,
2015-10-19T05:19:03.484Z,Internet Email,0802D52D6F874965,3,192.168.0.31:63715,64.233.189.27:25,>,EHLO EXSERVER.exchange2016demo.com,
2015-10-19T05:19:03.796Z,Internet Email,0802D52D6F874965,4,192.168.0.31:63715,64.233.189.27:25,<,"250 mx.google.com at your service, [203.206.161.219] SI
2015-10-19T05:19:03.796Z,Internet Email,0802D52D6F874965,5,192.168.0.31:63715,64.233.189.27:25,>,STARTTLS,
2015-10-19T05:19:04.109Z,Internet Email,0802D52D6F874965,6,192.168.0.31:63715,64.233.189.27:25,<,220 2.0.0 Ready to start TLS,
2015-10-19T05:19:04.124Z,Internet Email,0802D52D6F874965,7,192.168.0.31:63715,64.233.189.27:25,*,,CN=EXSERVER CN=EXSERVER 5253ED670317B8904C7A71F728E62E5
2015-10-19T05:19:04.765Z,Internet Email,0802D52D6F874965,8,192.168.0.31:63715,64.233.189.27:25,*,,CN=mx.google.com, O=Google Inc, L=Mountain View, S=Calif
2015-10-19T05:19:04.765Z,Internet Email,0802D52D6F874965,9,192.168.0.31:63715,64.233.189.27:25,*,,TLS protocol SP_PROT_TLS1_2_CLIENT negotiation succeeded
2015-10-19T05:19:04.765Z,Internet Email,0802D52D6F874965,10,192.168.0.31:63715,64.233.189.27:25,*,,F13AD248925B5CCCF9129BC9E3E0CA03F1DE74B1,Received certifi
2015-10-19T05:19:04.765Z,Internet Email,0802D52D6F874965,11,192.168.0.31:63715,64.233.189.27:25,>,EHLO EXSERVER.exchange2016demo.com,
2015-10-19T05:19:05.077Z,Internet Email,0802D52D6F874965,12,192.168.0.31:63715,64.233.189.27:25,<,"250 mx.google.com at your service, [203.206.161.219] SI
2015-10-19T05:19:05.077Z,Internet Email,0802D52D6F874965,13,192.168.0.31:63715,64.233.189.27:25,*,,sending message with RecordId 597000454159 and InternetM
2015-10-19T05:19:05.077Z,Internet Email,0802D52D6F874965,14,192.168.0.31:63715,64.233.189.27:25,>,MAIL FROM:<adam.wally@exchange2016demo.com> SIZE=5768,
2015-10-19T05:19:05.077Z,Internet Email,0802D52D6F874965,15,192.168.0.31:63715,64.233.189.27:25,>,RCPT TO:<exchangeserverpro@gmail.com>,
2015-10-19T05:19:05.394Z,Internet Email,0802D52D6F874965,16,192.168.0.31:63715,64.233.189.27:25,<,250 2.1.0 OK w14si49922774pbt.201 - gsmtpt,
2015-10-19T05:19:05.613Z,Internet Email,0802D52D6F874965,17,192.168.0.31:63715,64.233.189.27:25,<,250 2.1.5 OK w14si49922774pbt.201 - gsmtpt,
2015-10-19T05:19:05.628Z,Internet Email,0802D52D6F874965,18,192.168.0.31:63715,64.233.189.27:25,>,BDAT 3141 LAST,
2015-10-19T05:19:06.550Z,Internet Email,0802D52D6F874965,19,192.168.0.31:63715,64.233.189.27:25,<,250 2.0.0 OK w14si49922774pbt.201 - gsmtpt,
2015-10-19T05:19:06.550Z,Internet Email,0802D52D6F874965,20,192.168.0.31:63715,64.233.189.27:25,>,QUIT,
```

SMTP Relay for Applications and Devices

In most organizations there are several devices or applications that need to use an SMTP service to send email messages. An [Exchange 2016 server](#) can provide that service for you, however the configuration required on the server depends on the SMTP relay requirements of your scenario.

There are generally two types of SMTP relay scenarios that Exchange Server 2016 is used for:

- **Internal relay** – devices and applications that need to send email messages only to internal recipients in the Exchange organization.
- **External relay** – devices and applications that need to send email messages to external recipients.



Let's take a look at each of those scenarios, and then some additional considerations when you are deploying this in your own production environments.

Internal SMTP Relay

When Exchange Server 2016 is first installed the setup routine automatically creates a receive connector that is pre-configured to be used for receiving email messages from anonymous senders to internal recipients. This allows inbound internet email to be received by the server, and is also suitable for internal relay scenarios.

The receive connector is named "SERVERNAME\Default Frontend SERVERNAME", for example, "EXSERVER\Default Frontend EXSERVER" in my example environment.

```
[PS] C:\>Get-ReceiveConnector
```

| Identity | Bindings | Enabled |
|------------------------------------|---------------------------|---------|
| EXSERVER\Default Frontend EXSERVER | {0.0.0.0:2525, [::]:2525} | True |
| EXSERVER\Client Proxy EXSERVER | {[::]:465, 0.0.0.0:465} | True |

| | | |
|---|-------------------------|------|
| EXSERVER\Default Frontend EXSERVER | {[::]:25, 0.0.0.0:25} | True |
| EXSERVER\Outbound Proxy Frontend EXS... | {[::]:717, 0.0.0.0:717} | True |
| EXSERVER\Client Frontend EXSERVER | {[::]:587, 0.0.0.0:587} | True |

You can test this connector by [making an SMTP connection using Telnet](#) and issuing SMTP commands. For example:

```
C:\>telnet exserver 25

220 EXSERVER.exchange2016demo.com Microsoft ESMTP MAIL Service ready at Thu, 22
Oct 2015 11:39:23 +1000
helo
250 EXSERVER.exchange2016demo.com Hello [192.168.0.30]
mail from: test@test.com
250 2.1.0 Sender OK
rcpt to: adam.wally@exchange2016demo.com
250 2.1.5 Recipient OK
Data
354 Start mail input; end with .
Subject: Test email
Testing
.
250 2.6.0 <f7c2f921-ff7e-4ce4-b2eb-a70dc52f225f@EXSERVER.exchange2016demo.com> [
InternalId=854698491929, Hostname=EXSERVER.exchange2016demo.com] Queued mail for
delivery
```

So there's no specific configuration required on the server or the connectors to allow this scenario, however it is recommended that you use a DNS alias instead of the real server name. This will allow you to configure all of your devices and applications with the DNS alias, and you can later move that DNS alias to point to a different Exchange server during a migration.

External SMTP Relay

Continuing from the previous demonstration, let's see what happens if I try to use Telnet to send an email message from a valid internal address to an external recipient.

```
220 EXSERVER.exchange2016demo.com Microsoft ESMTP MAIL Service ready at Thu, 22
Oct 2015 12:04:45 +1000
helo
250 EXSERVER.exchange2016demo.com Hello [192.168.0.30]
mail from: adam.wally@exchange2016demo.com
250 2.1.0 Sender OK
rcpt to: exchangeserverpro@gmail.com
550 5.7.54 SMTP; Unable to relay recipient in non-accepted domain
```

An SMTP error code "550 5.7.54, Unable to relay recipient in non-accepted domain" is received instead. The receive connector will not allow an anonymous, unauthenticated sender to relay to external domain names, which prevents your server from being exploited as an open relay.

There are two ways you can resolve this and allow your devices and applications to send to external recipients:

- Using authentication for SMTP connections
- Configuring an anonymous SMTP relay connector

External SMTP Relay Using Authentication

The first method is to use authenticated SMTP connections. Exchange Server 2016 has a receive connector designed to be used by clients that need to send via SMTP called “SERVERNAME\Client Frontend SERVERNAME”, for example “EXSERVER\Client Frontend EXSERVER” in my example environment.

```
[PS] C:\>Get-ReceiveConnector
```

| Identity | Bindings | Enabled |
|---|---------------------------|---------|
| ----- | ----- | ----- |
| EXSERVER\Default EXSERVER | {0.0.0.0:2525, [::]:2525} | True |
| EXSERVER\Client Proxy EXSERVER | {[::]:465, 0.0.0.0:465} | True |
| EXSERVER\Default Frontend EXSERVER | {[::]:25, 0.0.0.0:25} | True |
| EXSERVER\Outbound Proxy Frontend EXS... | {[::]:717, 0.0.0.0:717} | True |
| EXSERVER\Client Frontend EXSERVER | {[::]:587, 0.0.0.0:587} | True |

Minimal configuration is required to get this working. Assuming you’ve already configured an SSL certificate for Exchange Server 2016, and added a DNS alias for your SMTP devices and applications to use (I’m using a DNS alias of **mail.exchange2016demo.com** in this example), you should then also set the **TlsCertificateName** for the receive connector.

Use Get-ExchangeCertificate to identify the thumbprint of the SSL certificate you’ll be using.

```
[PS] C:\>Get-ExchangeCertificate
```

| Thumbprint | Services | Subject |
|--|----------|--|
| ----- | ----- | ----- |
| FC5259C0528657EF22BB818CA9B23FD220A9DE83 | ...WS.. | CN=mail.exchange2016demo.com, OU=IT, O=LockLAN Systems Pty Ltd,... |
| FE6528BE1548D81C794AE9A00D144FF3D16E0CD2 |S.. | CN=Microsoft Exchange Server Auth Certificate |
| DAB089E53CA660DEF7B8EE303212C31C0E3D3499 | IP.WS.. | CN=EXSERVER |
| 17839AF62AA3A1CBB5F7EC81E92A609976D8AD9 | | CN=WMSvc-EXSERVER |

The syntax of the TlsCertificateName string is made up of two different attributes of the certificate, so I use the following commands to apply the configuration to my receive connector.

```
[PS] C:\>$cert = Get-ExchangeCertificate -Thumbprint FC5259C0528657EF22BB818CA9B23FD220A9DE83
```

```
[PS] C:\>$tlscertificatename = "<i>$(($cert.Issuer)<s>$(($cert.Subject))"
```

```
[PS] C:\>Set-ReceiveConnector "EXSERVER\Client Frontend EXSERVER" -Fqdn mail.exchange2016demo.com -TlsCertificateName $tlscertificatename
```

To test using the Client Frontend connector to send an email message I'm going to use PowerShell's [Send-MailMessage](#) cmdlet instead of Telnet. First, capture some valid credentials to use for authentication.

```
PS C:\>$credential = Get-Credential
```

Next, use the Send-MailMessage cmdlet with parameters specifying the server, to and from addresses, subject line, and the port number.

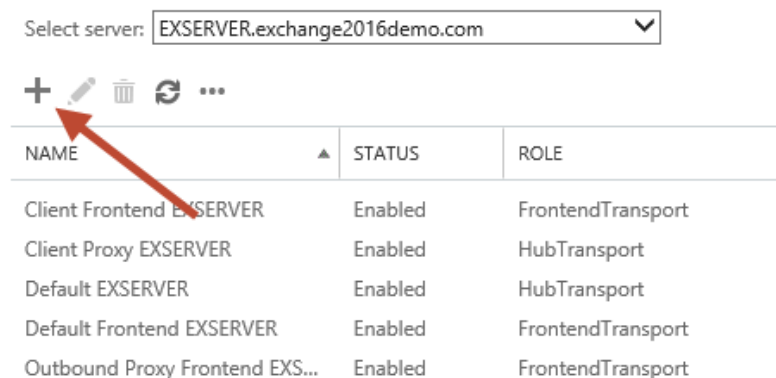
```
PS C:\>Send-MailMessage -SmtpServer mail.exchange2016demo.com -Credential $credential -From 'adam.wally@exchange2016demo.com' -To 'exchangeserverpro@gmail.com' -Subject 'Test email' -Port 587 -UseSsl
```

In the above example the email is successfully received by the external recipient. So any device or application on the network that can use authenticated SMTP can be set up to use that connector listening on port 587 on your Exchange 2016 server.

External SMTP Relay Using Anonymous Connections

When authenticated SMTP is not possible you can create a new receive connector on the Exchange 2016 server that will allow anonymous SMTP relay from a specific list of IP addresses or IP ranges.

In the Exchange Admin Center navigate to **mail flow** and then **receive connectors**. Select the server that you want to create the new receive connector on, and click the "+" button to start the wizard.



Give the new connector a name. I like to keep the name consistent with the other default connectors. Set the **Role** to “Frontend Transport”, and the **Type** to “Custom”.

*Name:
Anon Relay EXSERVER

Server:
EXSERVER.exchange2016demo.com

Role:
☐ Hub Transport
☒ Frontend Transport

Type:
☒ Custom (For example, to allow application relay)
☐ Internal (For example, to receive intranet mail)
☐ Internet (For example, to receive internet mail)
☐ Partner (For example, to route mail from trusted third-party servers)
☐ Client (For example, to receive mail from non-Outlook clients)

The default **Network adapter bindings** are fine. This represents the IP and port that the server will be *listening* on for connections. Multiple receive connectors on the Frontend Transport service can listen on the same port of TCP 25.

*Network adapter bindings:
Specify the IP addresses and port of the network adapter to bind to the receive connector.

+ -

| IP ADDRESSES | PORT |
|----------------------|------|
| (All available IPv4) | 25 |

Remove the default IP range from the **Remote network settings**, and then add in the specific IP addresses or IP ranges that you want to allow anonymous SMTP relay from. I do not recommend adding entire IP subnets that contain other Exchange servers as this can cause issues with server to server communications.

*Remote network settings:
Receive mail from servers that have these remote IP addresses.



| IP ADDRESSES |
|--------------|
| 192.168.0.30 |
| |

Click Finish to complete the wizard, then there is some additional configuration still required.

In the Exchange Management Shell run the following two commands.

```
[PS] C:\>Set-ReceiveConnector "EXSERVER\Anon Relay EXSERVER" -PermissionGroups AnonymousUsers  
[PS] C:\>Get-ReceiveConnector "EXSERVER\Anon Relay EXSERVER" | Add-ADPermission -User 'NT  
AUTHORITY\Anonymous Logon' -ExtendedRights MS-Exch-SMTP-Accept-Any-Recipient
```

We can now test the connector using Telnet from the IP address that was added to the remote network settings of the receive connector. In my test environment that IP address will now be allowed to send email from any email address (whether it is a valid internal address or not) to any external address.

```
220 EXSERVER.exchange2016demo.com Microsoft ESMTP MAIL Service ready at Thu, 22  
Oct 2015 12:59:39 +1000  
helo  
250 EXSERVER.exchange2016demo.com Hello [192.168.0.30]  
mail from: test@test.com  
250 2.1.0 Sender OK  
rcpt to: exchangeserverpro@gmail.com  
250 2.1.5 Recipient OK  
Data  
354 Start mail input; end with .  
Subject: test  
.  
250 2.6.0 <e1739c5f-db11-4fdd-aa27-a9702bc15b15@EXSERVER.exchange2016demo.com> [  
InternalId=863288426497, Hostname=EXSERVER.exchange2016demo.com] Queued mail for  
delivery
```

Extra Considerations for SMTP Relay

Here's some additional items that you should consider when you're providing SMTP relay services with Exchange Server 2016 for your environment.

A lot of organizations simply go with the anonymous relay option and set up a connector that allows wide ranges of IP addresses to relay email anywhere. This is the simplest approach, but

clearly not the best in terms of security and auditing. Anonymous relay relies on trusted, identifiable IP addresses. If the IP addresses are in a DHCP pool, are associated with a load balancer (see above), are multi-user (such as terminal servers), or the IP/host itself is compromised in some way, then your ability to trace emails back to the real source is difficult if not impossible.

Although authentication adds some complexity, it may be worth it from security perspective. However, it does mean managing credentials for all of your devices and applications. Sharing SMTP credentials across multiple systems might seem like a way to avoid complexity, but it re-introduces the problems associated with anonymous SMTP.

In the example above I demonstrated configuring a TLS certificate name for a receive connector and also used TLS/SSL for my testing with Send-MailMessage. Although unencrypted SMTP may seem easier, if you are going to use authentication for SMTP in your environment, or the SMTP traffic is in any way sensitive, then you should protect it with TLS/SSL encryption so that the credentials can't be compromised by network attackers.

You may also be wondering how the Exchange server is able to differentiate between traffic destined for one receive connector vs another receive connector, when both of them are listening on the same IP address and port number, for example "EXSERVER\Default Frontend EXSERVER" and "EXSERVER\Anon Relay EXSERVER".

The answer is in the **Remote network settings** of the receive connectors. Exchange will use the receive connector that is the *most specific match* for the source IP address of the SMTP connection.

In my examples above this means that the default connector with its remote network settings of 0.0.0.0-255.255.255.255 (which is basically "anywhere") is less specific than the relay connector with its remote network settings of 192.168.0.30. So when an SMTP connection comes from IP 192.168.0.30 to port 25 on the server it will be handled by the relay connector, while everything else connecting to port 25 will be handled by the default connector.

Configuring Mailbox

Mailbox services refers to the databases that host the different types of mailboxes such as user mailboxes, shared mailboxes, and public folder mailboxes. For a newly installed Exchange 2016 server there are a few tasks required for configuring mailbox databases.

Moving a Mailbox Database

When Exchange Server 2016 is installed it creates a mailbox database for you on the server. If you installed Exchange to the default path, then the mailbox will be stored in **C:\Program Files\Microsoft\Exchange Server\V15\Mailbox**.

Here's an example from my example server:

```
[PS] C:\>Get-MailboxDatabase | fl Name,EdbFilePath,LogFolderPath  
  
Name           : Mailbox Database 2116642217  
EdbFilePath    : C:\Program Files\Microsoft\Exchange Server\V15\Mailbox\Mailbox Database  
                2116642217\Mailbox Database 2116642217.edb  
LogFolderPath  : C:\Program Files\Microsoft\Exchange Server\V15\Mailbox\Mailbox Database  
                2116642217
```

The default location is probably not suitable for your environment, so you would likely want to move this database to the volumes that you've provisioned for your Exchange database and log files. Another common scenario is that the database is growing and the current volume is low on free disk space, so you want to move it to new, larger volume.

A mailbox database can be moved, but before you proceed consider that the move requires the database to be dismounted and taken offline, so it will not be accessible by your mailbox users during the move. This is fine if the server has just been set up and there are no mailboxes on it, but if you're moving a database with active mailbox users a better option would be to create a new database on the new volume and perform mailbox moves (which are non-disruptive to end users).

I will also point out that the procedure below is not suitable for mailbox databases that are being replicated to multiple DAG members.

Before I move the database I am first going to rename it. The uniquely generated name of "Mailbox Database 2116642217" is not desirable. After all, we're going to be typing database

names when running PowerShell commands, so it should be a shorter name that is easier to remember. So I will rename it to “DB01” instead.

```
[PS] C:\>Set-MailboxDatabase "Mailbox Database 2116642217" -Name "DB01"
```

To move the database and transaction log files to their new locations we use the Move-DatabasePath cmdlet.

```
[PS] C:\>Move-DatabasePath DB01 -EdbFilePath D:\DB01\DB01.edb -LogFolderPath E:\DB01
```

The database is temporarily dismounted, the files are copied to the new locations, and then the database is mounted again. The time the operation takes will depend on how much data there is to be moved, as well as the speed of the source and destination disks. Generally speaking, the more data you have the longer it will take, potentially becoming a very long outage for the database. On a new database that isn’t hosting any mailboxes yet it will be very quick, and there’s no users that would be impacted anyway. But for a database that is hosting users I recommend considering moving mailboxes to a new database instead.

Creating a New Mailbox Database

Most organizations will need more than one mailbox database. Although a single database can be several terabytes in size (up to 16TB is technically supported) it is recommended to keep the database sizes smaller.

In particular, if you’re running just a single Exchange 2016 server it is recommended to keep the databases to less than 200GB each. Consider a scenario where you need to restore a database from backup so that you can recover one mailbox. Restoring 200GB of data will be much faster, and easier to find disk space for, than restoring 1TB of data.

Remember, a Standard Edition Exchange 2016 server can host up to 5 databases, so you might as well make use of it.

To create a new mailbox database we use the New-MailboxDatabase cmdlet. Here’s an example:

```
[PS] C:\>New-MailboxDatabase -Server EXSERVER -Name DB02 -EdbFilePath D:\DB02\DB02.edb -  
LogFolderPath E:\DB02
```

| Name | Server | Recovery | ReplicationType |
|------|----------|----------|-----------------|
| ---- | ----- | ----- | ----- |
| DB02 | EXSERVER | False | None |

```
WARNING: Please restart the Microsoft Exchange Information Store service on server EXSERVER after adding new mailbox databases.
```

```
[PS] C:\>Mount-Database DB02
```

Note that there's a warning that advises us to restart the information store service. This is recommended because the information store service allocates memory to each mailbox database when it starts up, and it bases that allocation on the number of databases on the server. When you create a new mailbox database an information store restart is required to re-allocate the memory equally to each database on the server.

Although it's not critical that you immediately perform that service restart, I do recommend you do it before you move or create any mailboxes on the new database. A service restart outside of business hours or during an approved outage windows will take care of it.

```
[PS] C:\>Restart-Service MExchangeIS
WARNING: Waiting for service 'Microsoft Exchange Information Store (MExchangeIS)' to stop...
WARNING: Waiting for service 'Microsoft Exchange Information Store (MExchangeIS)' to start...
```

Configuring Mailbox Database Quotas

The mailbox databases on an Exchange 2016 server are pre-configured with mailbox quotas that will be applied to the mailboxes that the databases host. The default quotas are fairly small for what the typical email user needs these days. Of course, new users start out with 0GB of mailbox storage consumed, and might take a year or so to reach the default quota levels, but if you're dealing with existing mailbox users there's a good chance they've already got more mailbox data than the default quotas allow.

To view the current mailbox quota levels use the Get-MailboxDatabase cmdlet.

```
[PS] C:\>Get-MailboxDatabase | fl Name,IssueWarning*,ProhibitSend*
```

```
Name                : DB01
IssueWarningQuota    : 1.899 GB (2,039,480,320 bytes)
ProhibitSendReceiveQuota : 2.3 GB (2,469,396,480 bytes)
ProhibitSendQuota     : 2 GB (2,147,483,648 bytes)

Name                : DB02
IssueWarningQuota    : 1.899 GB (2,039,480,320 bytes)
ProhibitSendReceiveQuota : 2.3 GB (2,469,396,480 bytes)
ProhibitSendQuota     : 2 GB (2,147,483,648 bytes)
```

As you can see the default quota is around 2GB.

Let's increase this to a more generous level of 10GB for this example.

```
[PS] C:\>Get-MailboxDatabase | Set-MailboxDatabase -IssueWarningQuota 9.5GB -ProhibitSendQuota 10GB  
-ProhibitSendReceiveQuota 12GB
```

The outcome will be:

- When a mailbox reaches 9.5GB in size the user will receive a warning notification in Outlook and in their inbox.
- When a mailbox reaches 10GB in size the user will be unable to send new email messages, but will still be able to receive email.
- When a mailbox reaches 12GB in size the user will be unable to send new mail messages, and any email sent to the mailbox will be rejected with a non-delivery report sent back to the original sender.

I always recommend configuring a "ProhibitSendReceiveQuota" so that mailboxes can't grow endlessly, for example if a user leaves the organization but their mailbox is still receiving email from distribution lists or external parties. Unrestricted growth will cause you a lot of problems when one day you discover a 100GB mailbox has been quietly growing on your databases.

There's one more setting you should check for your databases.

```
[PS] C:\>Get-MailboxDatabase | fl Name,Retain*,*retention
```

```
Name : DB01  
RetainDeletedItemsUntilBackup : False  
MailboxRetention : 30.00:00:00  
DeletedItemRetention : 14.00:00:00
```

```
Name : DB02  
RetainDeletedItemsUntilBackup : False  
MailboxRetention : 30.00:00:00  
DeletedItemRetention : 14.00:00:00
```

By default, a mailbox database will retain delete items for 14 days, and deleted mailboxes for 30 days. While they are being retained you can recover accidentally deleted items or mailboxes without needing to restore your database from a backup. After those thresholds are passed the deleted items or mailboxes are purged from the database.

Also by default, a mailbox database will purge the retained items and mailboxes whether a backup has run recently or not. This makes sense for Microsoft's Office 365 service, where no traditional backups are run. But in your on-premises deployment where you're running regular backups, this default setting is not desirable. If you have failing backups it's possible that an

accidentally deleted item or mailbox will be purged from the database before you've had a successful backup of it, making recovery impossible. So I recommend you change the retention setting for your new mailbox databases.

```
[PS] C:\>Get-MailboxDatabase | Set-MailboxDatabase -RetainDeletedItemsUntilBackup $true
```

Managing Recipients

Now that we've got an Exchange 2016 server installed and configured, it's time to look at managing recipients. Let's face it, installing the server is a one-time task, and for a lot of administrators they'll only install that single server and not do it again until several years later when they upgrade to the next version of Exchange Server.

Recipient management, on the other hand, is a task that you'll be performing on a daily basis. Let's take a look at some common recipient management tasks in our example environment.

Creating Mailboxes

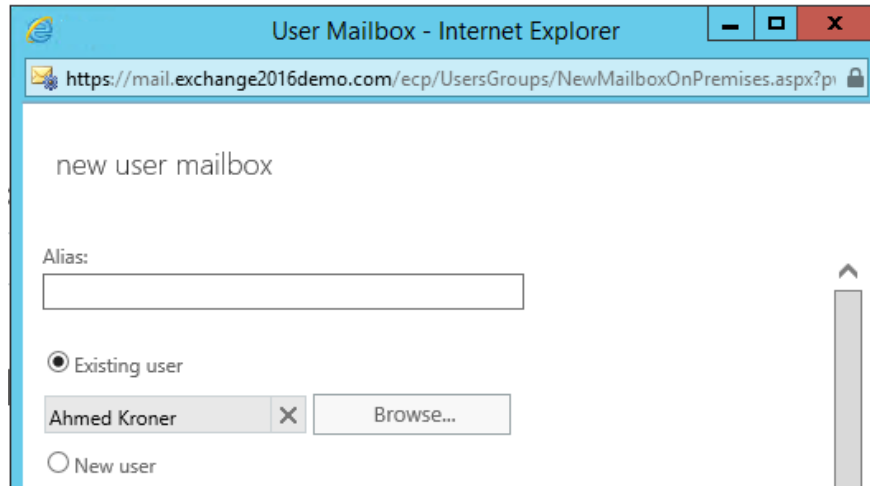
Exchange Server 2016 is tightly integrated with Active Directory. Every mailbox that you create in Exchange is associated with an Active Directory user object. In fact, when you create a mailbox what you're actually doing is mailbox-enabling an Active Directory user.

Earlier in this eBook we created a new user and mailbox using the Exchange Admin Center. Let's look at two other mailbox creation scenarios.

Before you do this in your own test lab you should create some Active Directory users first. You can create them manually if you like, or you can use my [New-LabUsers.ps1](#) PowerShell script to automatically create them for you.

Mailbox-Enabling a User

We can create a mailbox for an existing Active Directory user using the Exchange Admin Center. Similar to the last time, we use the **recipients → mailboxes** section of the EAC, but this time choose an existing user, and click **Save** to create the mailbox.



To do the same task in the Exchange Management Shell we can use `Get-User` and `Enable-Mailbox` together.

```
[PS] C:\>Get-User Ahmed.Kroner | Enable-Mailbox
```

| Name | Alias | ServerName | ProhibitSendQuota |
|--------------|--------------|------------|-------------------|
| ---- | ----- | ----- | ----- |
| Ahmed Kroner | Ahmed.Kroner | exserver | Unlimited |

That's nice and easy, but for your test lab you might like to create dozens or even hundreds of mailboxes to perform your testing. So let's take a look at bulk creation of mailboxes.

Creating Multiple Mailboxes

So far we've been creating individual mailboxes, and allowing Exchange to automatically choose which database to place them on. Using PowerShell, we can see how many mailboxes are on each database right now.

```
[PS] C:\>Get-Mailbox | Group-Object -Property:Database | Select Name,Count | ft -auto
```

| Name | Count |
|------|-------|
| ---- | ----- |
| DB01 | 3 |
| DB02 | 2 |

Exchange will automatically balance the number of mailboxes hosted on each available database so that they have a reasonably balanced load, although it only takes into account the number of mailboxes, not the actual size of them.

If we create multiple mailboxes in a single operation, we should see those numbers stay pretty close. For this example, I'm going to mailbox-enable every user in my Company/Users OU in Active Directory.

```
[PS] C:\>Get-User -OrganizationalUnit "ou=users,ou=company,dc=exchange2016demo,dc=com" | Enable-Mailbox
```

When you run that command in your own test lab environment you may see some errors if any of the users already have mailboxes, which is nothing to be concerned about.

Now let's take another look at the mailbox distribution for each database.

```
[PS] C:\>Get-Mailbox | Group-Object -Property:Database | Select Name,Count | ft -auto
```

| Name | Count |
|------|-------|
| DB01 | 52 |
| DB02 | 51 |

As you can see Exchange does a pretty good job of balancing the user load across databases. You may still need to move mailboxes around from time to time, so that the growth of each database stays consistent with the others. But as an administrator you don't need to worry about choosing a database every time you create a new mailbox. You can simply allow Exchange to automatically select one for you.

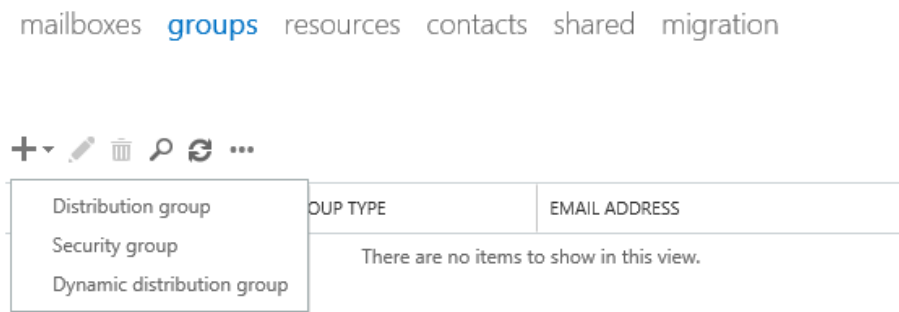
Creating Distribution Groups

Distribution groups are used to send email to multiple recipients at once. Similar to the relationship between Active Directory users and mailboxes, a distribution group is simply a mail-enabled group in Active Directory.

The Active Directory group can be either a Security group or a Distribution group, which can be confusing at first. "Distribution group" is a type of group in Active Directory, but the group can't be used to send email until you have mail-enabled it in Exchange. "Security group" is the other type of group in Active Directory. You can mail-enable security groups, which is useful when need the group to be used for both email distribution as well as granting access to a resource (such as a shared mailbox or a file share).

Whether you're using security or distribution groups the only requirement is that they are Universal in scope, so that Exchange can see them.

Logon to the Exchange Admin Center and navigate to **recipients** → **groups**. Create a new group, noting that there is a third type of group called “Dynamic distribution group” (DDG).



DDGs are dynamically populated based on a query, for example you can create a DDG that includes all mailbox-enabled users with a Department attribute of “Human Resources”. An advantage of DDGs is that you don’t need to manage the membership, it happens automatically as users attributes are modified.

For now, let’s create a new distribution group that will be used to send email to all staff in the organization.

new distribution group

*Display name:
All Staff

*Alias:
allstaff

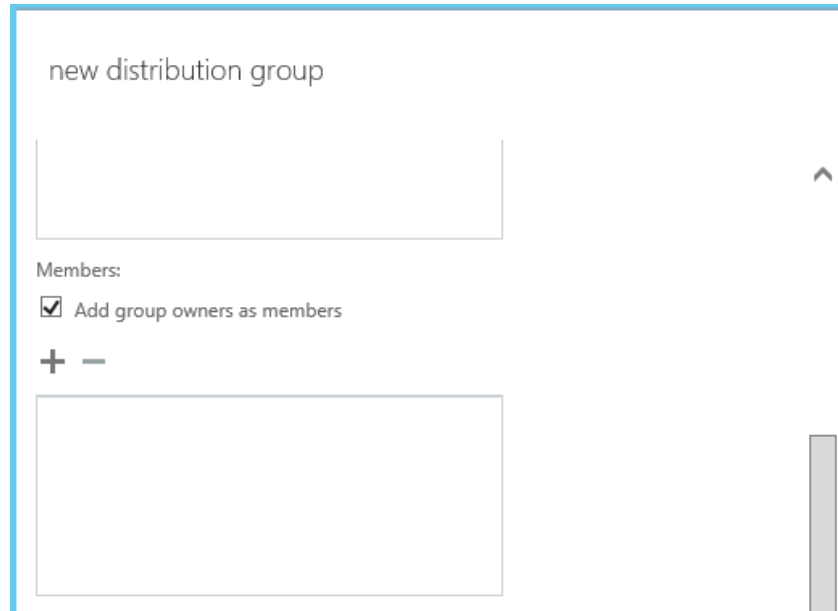
Notes:

Organizational unit:
exchange2016demo.com/C X Browse...

*Owners:
+ -

| |
|---------------|
| Administrator |
|---------------|

Scroll down the new distribution group form until you see the Members section. Here you can add members to the group at the same time as you are creating it. I'm going to leave this blank for now, and populate the group using PowerShell instead.



After creating the group let's go back to the Exchange Management Shell. We want to add all mailbox-enabled users in the "Users" OU to the distribution group, which we can do by running Get-Mailbox and Add-DistributionGroupMember. The reason I'm specifying the OU here is so that other mailboxes that may exist in other OUs, such as an OU containing shared mailboxes, are not also added to the distribution group.

```
[PS] C:\>Get-Mailbox -OrganizationalUnit "ou=users,ou=company,dc=exchange2016demo,dc=com" | Add-DistributionGroupMember "All Staff"
```

Managing Distribution Groups

Now let's take a look at how to manage existing distribution groups. There are a few important settings to be aware of for the distribution groups in your organization, especially those that control who can send to the distribution group.

From the Exchange Admin Center open the properties of a distribution group, such as the "All Staff" group that was created earlier.

In the **delivery management** settings notice that the default setting for a new distribution group is to only allow senders inside the organization to send to it.

All Staff

general
ownership
membership
membership approval
▸ **delivery management**
message approval
email options
MailTip
group delegation

By default, only senders inside your organization can send messages to this group. You can also allow people outside the organization to send to this group. Choose one of the options below.

- ☒ Only senders inside my organization
☐ Senders inside and outside of my organization

If you want to restrict who can send messages to the group, add users or groups to the list below. Only the specified senders will be able to send to the group and mail sent by anyone else will be rejected.

+ -

All senders can send messages to this group.

On the other hand, if you have a distribution group for your sales team then it is likely that they will want to receive external email messages from customers. In that case, after creating the distribution group for the sales team you would modify the delivery management settings to allow outside senders.

Managing Delivery Restrictions for Distribution Groups

The default setting shown above ensures that your groups are not inadvertently exposed to abuse from external senders. Sensitive groups, or very large groups, will often be left with this configuration in place.

However, there's also valid reasons to secure distribution groups from abuse by internal senders. For example, in a large organization the "All Staff" distribution group will generate a high volume of email traffic, and this can cause problems when end users start using "Reply all" in Outlook.

A "reply all storm" as more and more people start replying (usually to complain about the people using "reply all") can quickly overwhelm an Exchange server and cause it to perform very badly, if not crash entirely.

So for high impact or sensitive distribution groups you can lock them down even further by specifying who can send to it. In this example I'm restricting the "All Staff" group so that only the CEO's administrative team can send to it.

By default, only senders inside your organization can send messages to this group. You can also allow people outside the organization to send to this group. Choose one of the options below.

☒ Only senders inside my organization

☐ Senders inside and outside of my organization

If you want to restrict who can send messages to the group, add users or groups to the list below. Only the specified senders will be able to send to the group and mail sent by anyone else will be rejected.

+ -

| |
|---------------|
| Beverlee Lowy |
| Dillon Blowe |
| Graham Layton |
| |

When this list is empty, group members will receive messages from everyone who sends to this group.

If you add senders to this list, group members will receive messages only from those senders and reject messages from everyone else.

Using Moderation for Distribution Groups

Sometimes a large distribution group is intended for use by everyone in the organization, but you still want to maintain some control over the email messages that are sent to it. This is where moderation comes into play.

In the **message approval** settings of the distribution group enable moderation and specify one or more moderators for the group. Moderators will receive an email in their inbox to review all messages sent to the distribution group, and they can then click a button to approve or reject the message.

Backup and Recovery

Now that we've got the Exchange 2016 server fully configured and populated with some mailboxes and groups it's time to turn our attention to protecting all of that work.

Backing up your Exchange server is, of course, a very important task. Over the years I have witnessed many unfortunate data loss incidents that were ultimately the fault of incorrectly configured, or non-operational backups. The simple fact is that failures happen. One day your server or storage will fail, and you'll need to recover the data that it was hosting. You should *expect that to happen*. If you don't have reliable backups, then you might find yourself suddenly needing to update your resume and get your best suit dry cleaned.

Let's get started by covering some of the general concepts around Exchange Server 2016 backup and recovery.

Backup and Recovery Terminology

As you deal with different Exchange Server backup and recovery scenarios you'll encounter a lot of the same terminology, so let's start with that.

Types of Backup

There are four backup types that you'll generally see referred to in backup products and documentation.

- **Full** – a complete copy of the data on a server, volume, application or file system. For Exchange 2016 database backups a “full” backup is sometimes also referred to as a “VSS Full” or an “application aware” backup. A full backup will include all of the data regardless of whether the data has changed since the last backup or not. A full backup is a complete set of data that can be used for a restore.
- **Incremental** – a partial copy of the data on a server, volume, application or file system. An incremental backup will only include data that has changed since the last full or incremental backup. In a restore scenario the last full backup, plus all subsequent incremental backups up to the point in time you're restoring to, will be required for the recovery to be successful.

- **Differential** – similar to an incremental backup, however a differential backup does not mark the data as having been backed up. This means that differential backup sets tend to get larger and larger as you get further away from the last full backup. However, in a restore scenario differentials can be simpler than incrementals because you only need the last full backup plus the *latest* differential backup to perform the recovery.
- **Copy** – similar to a full, however the data is not marked as having been backed up. Copy backups are typically used to make a copy of data to another system for testing purposes. Copy backups are not suitable for recovery scenarios involving Exchange databases.

Each backup type has pros and cons. Full backups are the simplest to operate and recover from, but take the longest to run. Using a mixture of full backups plus incremental or differential backups can shorten some of your backup job times, but at the cost of extra time and complexity when you need to perform a recovery.

In addition to the backup types listed above you'll encounter other terminology in various backup products such as "synthetic full" backups. Those terms can mean many different things depending on the backup vendor, so you should refer to the specific documentation for those products to find out more.

Backup Storage

Different backup products support writing and storing backup sets on a wide variety of storage types.

- **Tape** – magnetic tape backup media that is available in many different formats and capacities. Tape is still commonly used today but not always as a primary backup media. Instead it is often used to replicate backup sets from disk storage so that a copy of the backed up data can be taken offsite.
- **Disk** – very large capacity disk storage is very cheap these days, faster than tape for many backup and recovery scenarios, and often has attractive features such as hardware-based de-duplication, compression, and replication.
- **Cloud** – there are many cloud-based storage providers to choose from these days, such as Amazon Web Services and Microsoft Azure. These providers sell storage by the gigabyte, usually at very low cost. Cloud-based backup storage often includes built-in replication of your data to protect it from failures in the cloud provider's infrastructure.

Cloud-based backup storage is also very practical in that you do not need to purchase large amounts of it up front as you do with on-site backup storage.

Cloud-based backup is becoming very popular these days, however backing up large amounts of data to the cloud does require good network bandwidth between you and your provider. It can also be slower to restore from. Some organizations use cloud-based storage as an off-site replica of their on-premises disk-based backup storage. Some even combine all three, backing up primarily to on-site disk, then replicating that to the cloud while also making copies of specific data to tape (usually multiple tapes) to be stored off-site for specific retention requirements.

When you are considering backup storage for your Exchange 2016 servers remember to follow the 3-2-1 Rule:

- At least **3** copies of the data
- Stored on at least **2** different media
- At least **1** copy kept off-site

Other General Terminology

Here's some additional terms you may need to be familiar with.

- **RPO** – stands for Recovery Point Objective. The RPO is the point in time that you are attempting to recover data from. For example, attempting to recover a mailbox from 5pm last Monday. The RPO may also define how much data loss a business is willing to accept in the event of a disaster, and your backup solution should be designed to meet that RPO. For example, if the business tells you that they are willing to accept up to 24 hours of data loss, then running only a weekly backup is obviously not acceptable.
- **RTO** – stands for Recovery Time Objective. The RTO defines the amount of time that is acceptable to perform a recovery after a disaster. Your backup solution should be designed to meet the RTO as well. For example, if the business requires an RTO of 8 hours but it would take you 20 hours to retrieve tapes from off-site storage and recover from them, then you would not be able to meet the RTO. However, you should also be aware that the RTO can be impacted by infrastructure other than the Exchange server itself. If you virtualize your Exchange servers the the virtualization hosts are lost in a

disaster, then obviously you can't start to recover the Exchange VM until some other host is available.

- **VSS** – stands for Volume Shadow-copy Service. VSS is part of the Windows Server operating system and is used to make application-aware backups of Exchange 2016 databases.
- **Recovery Database** – a special type of Exchange server database that is used as the target for a database restore operation. Data within the mailboxes of a recovery database can't be accessed by clients but can be extracted by the administrator and restored into a user's mailbox.
- **Database Portability** – this refers to Exchange Server 2016's capability to mount databases that have been copied or restored from other Exchange 2016 servers within the same Exchange organization. This is useful when the original server that hosted the database is no longer available for the recovery operation.
- **Dial Tone Recovery** – this refers to Exchange Server 2016's capability to mount a temporary database with empty mailboxes for end users to connect to so that they can continue to send and receive emails. A dial tone recovery is often used to restore *service* for end users while the much longer process of recovery the mailbox data from backup is performed.
- **Log Truncation** – all changes (transactions) to an Exchange 2016 database are stored in a memory buffer and also written to transaction log files. Periodically the memory buffer is flushed by committing changes to the database file itself. As there is generally some gap between what is written to the transaction log files and what has been committed to the database the log files become very important in a recovery scenario. Transaction logs accumulate on the server (and consume disk space) until the next database backup. When a full backup of the database is taken the server will remove the transaction log files that are no longer needed for recovery now that a backup of the database up to that specific point in time has been successfully taken.
- **Circular Logging** – when circular logging is enabled the transaction logs are automatically truncated as the changes are committed to the database file. This reduces the disk space consumption by the transaction logs, but removes the ability to recover the database beyond the point of the most recent backup.

As an additional note you may encounter snapshot-based backup systems in the real world, especially when you're running virtualized Exchange servers. While a snapshot-based backup

solution may still be supported for backups, providing it takes an application-aware backup that properly truncates the transaction logs, **snapshots are not supported for recovery purposes**.

Many snapshot-based backup products provide different processes or tools to use for recovering data from their backup sets that do not involve “rolling back” the VM to the last snapshot, which is fine. I mention this because a common mistake by administrators is to take a snapshot of an Exchange VM before any routine maintenance (such as monthly security patching) with the expectation that they can “roll back” the VM using that snapshot if something goes wrong with the patching. Unfortunately, this type of snapshot recovery can be catastrophic for an Exchange server.

What to Back Up for Exchange Server 2016

Exchange Server 2016 has two server roles; Mailbox and Edge Transport. The backup requirements for each server role are different.

- **Edge Transport** – a full server backup is generally advisable; however, it is not necessarily a requirement. If your ability to rebuild and reinstall the Edge Transport server (for example with an automated operating system deployment and pre-scripted Exchange installation and configuration) allows you to restore operation within an acceptable timeframe, then you would not necessarily need to also use traditional backups for the server.
- **Mailbox** – similar to the Edge Transport server you may consider not backing up the server operating system itself if you have fast enough rebuild processes. However, Mailbox servers also host the databases containing mailbox and public folder data, as well as the transport databases that may contain email messages still in transit. Therefore, it is recommended to back up at least the databases, if not the entire server.

Aside from the considerations above you should also think about the various log files that are stored on the Exchange servers, such as event logs or message tracking logs. Those are important for historical purposes.

If you’re in any doubt as to what you should be backing up on your Exchange servers, I recommend you err on the side of caution and backup everything.

Backing Up Exchange Server 2016 using Windows Server Backup

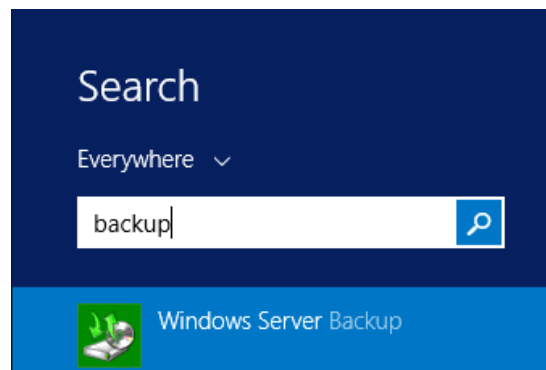
Windows Server Backup is a free backup utility provided with Windows Server. It is compatible with Exchange Server 2016, and although it is not as feature-rich as commercial backup products it does the job of protecting your Exchange data. And for a test lab environment it is the perfect way to learn the ins and outs of Exchange Server 2016 backup and recovery.

The Windows Server Backup feature is not installed by default so the first step we need to perform is to install it on the server.

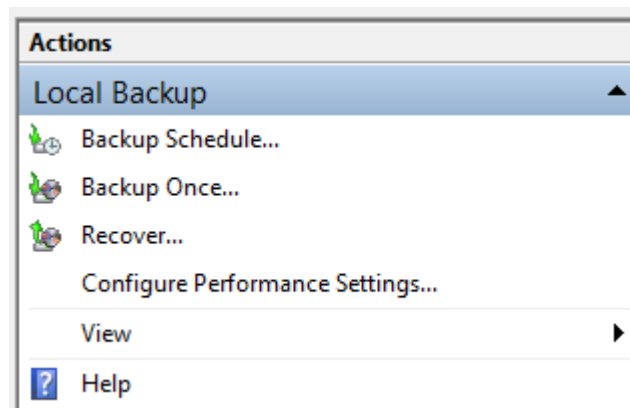
```
PS C:\> Install-WindowsFeature Windows-Server-Backup
```

| Success | Restart Needed | Exit Code | Feature Result |
|---------|----------------|-----------|-------------------------|
| True | No | Success | {Windows Server Backup} |

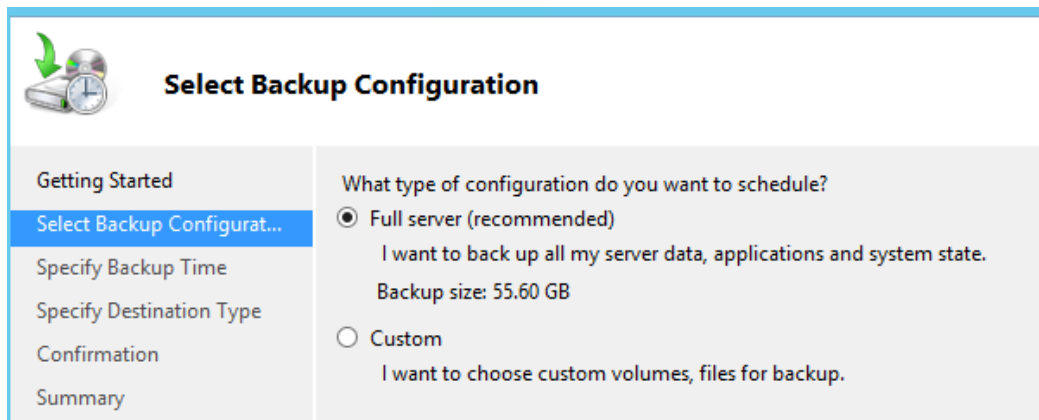
Now we can launch Windows Server Backup and configure the scheduled backup job.



After launching Windows Server Backup select **Local Backup** from the left side and then click **Backup Schedule** in the Actions pane on the right side.



When the Backup Schedule Wizard launches you have the choice to create a full server backup, or a custom backup.



The screenshot shows the 'Select Backup Configuration' step of the Backup Schedule Wizard. On the left is a navigation pane with the following items: 'Getting Started', 'Select Backup Configurat...', 'Specify Backup Time', 'Specify Destination Type', 'Confirmation', and 'Summary'. The 'Select Backup Configurat...' item is highlighted. The main area is titled 'Select Backup Configuration' and contains the question 'What type of configuration do you want to schedule?'. There are two radio button options: 'Full server (recommended)' and 'Custom'. The 'Full server (recommended)' option is selected. Below it, the text reads 'I want to back up all my server data, applications and system state.' and 'Backup size: 55.60 GB'. The 'Custom' option is also visible with the text 'I want to choose custom volumes, files for backup.'

Select Backup Configuration

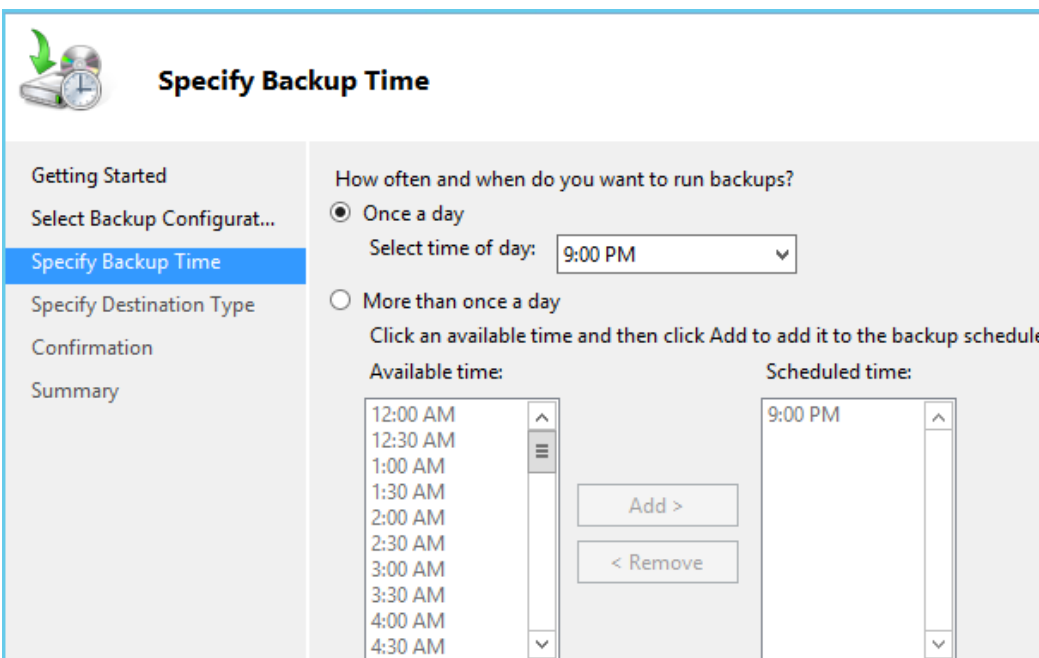
Getting Started
Select Backup Configurat...
Specify Backup Time
Specify Destination Type
Confirmation
Summary

What type of configuration do you want to schedule?

☒ Full server (recommended)
I want to back up all my server data, applications and system state.
Backup size: 55.60 GB

☐ Custom
I want to choose custom volumes, files for backup.

Select **Full server** from the backup configuration choice and then click Next. Choose the backup time and frequency you need for your environment.



The screenshot shows the 'Specify Backup Time' step of the Backup Schedule Wizard. The navigation pane on the left is the same as in the previous screenshot, but 'Specify Backup Time' is now highlighted. The main area is titled 'Specify Backup Time' and contains the question 'How often and when do you want to run backups?'. There are two radio button options: 'Once a day' and 'More than once a day'. The 'Once a day' option is selected. Below it, there is a 'Select time of day:' label and a dropdown menu showing '9:00 PM'. The 'More than once a day' option is also visible with the text 'Click an available time and then click Add to add it to the backup schedule.' Below this, there are two columns: 'Available time:' and 'Scheduled time:'. The 'Available time:' column has a list of times from 12:00 AM to 4:30 AM in 30-minute increments. The 'Scheduled time:' column has a single time slot showing '9:00 PM'. Between the two columns are two buttons: 'Add >' and '< Remove'.

Specify Backup Time

Getting Started
Select Backup Configurat...
Specify Backup Time
Specify Destination Type
Confirmation
Summary

How often and when do you want to run backups?

☒ Once a day
Select time of day: 9:00 PM

☐ More than once a day
Click an available time and then click Add to add it to the backup schedule.

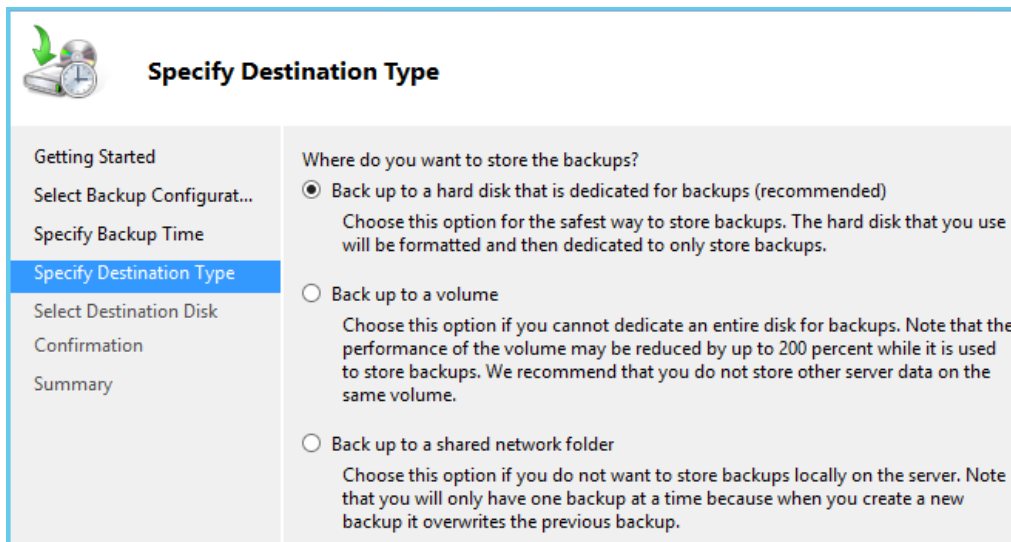
Available time: Scheduled time:

12:00 AM
12:30 AM
1:00 AM
1:30 AM
2:00 AM
2:30 AM
3:00 AM
3:30 AM
4:00 AM
4:30 AM

9:00 PM

Add >
< Remove

Choose your backup destination. I'm backing up to a hard disk that is dedicated for Windows Server Backup to use.



The 'Specify Destination Type' screen features a left-hand navigation pane with the following steps: Getting Started, Select Backup Configurat..., Specify Backup Time, Specify Destination Type (highlighted), Select Destination Disk, Confirmation, and Summary. The main area is titled 'Specify Destination Type' and contains the question 'Where do you want to store the backups?'. It offers three radio button options: 'Back up to a hard disk that is dedicated for backups (recommended)' (selected), 'Back up to a volume', and 'Back up to a shared network folder'. Each option includes a descriptive paragraph explaining its use and limitations.

Specify Destination Type

Getting Started
Select Backup Configurat...
Specify Backup Time
Specify Destination Type
Select Destination Disk
Confirmation
Summary

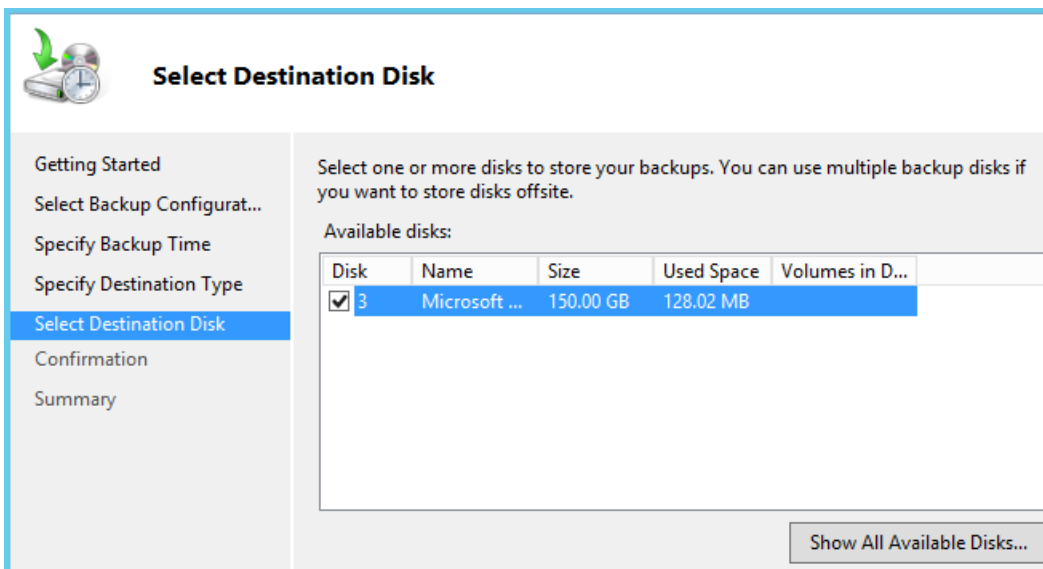
Where do you want to store the backups?

☒ Back up to a hard disk that is dedicated for backups (recommended)
Choose this option for the safest way to store backups. The hard disk that you use will be formatted and then dedicated to only store backups.

☐ Back up to a volume
Choose this option if you cannot dedicate an entire disk for backups. Note that the performance of the volume may be reduced by up to 200 percent while it is used to store backups. We recommend that you do not store other server data on the same volume.

☐ Back up to a shared network folder
Choose this option if you do not want to store backups locally on the server. Note that you will only have one backup at a time because when you create a new backup it overwrites the previous backup.

Click **Show All Available Disks** and add the volume you're using for backups, then select it by ticking the box before you continue to the next step.



The 'Select Destination Disk' screen has a left-hand navigation pane with steps: Getting Started, Select Backup Configurat..., Specify Backup Time, Specify Destination Type, Select Destination Disk (highlighted), Confirmation, and Summary. The main area is titled 'Select Destination Disk' and includes the instruction 'Select one or more disks to store your backups. You can use multiple backup disks if you want to store disks offsite.' Below this is a table of 'Available disks'. The first row is selected, with a checkbox checked. A 'Show All Available Disks...' button is located at the bottom right.

Select Destination Disk

Getting Started
Select Backup Configurat...
Specify Backup Time
Specify Destination Type
Select Destination Disk
Confirmation
Summary

Select one or more disks to store your backups. You can use multiple backup disks if you want to store disks offsite.

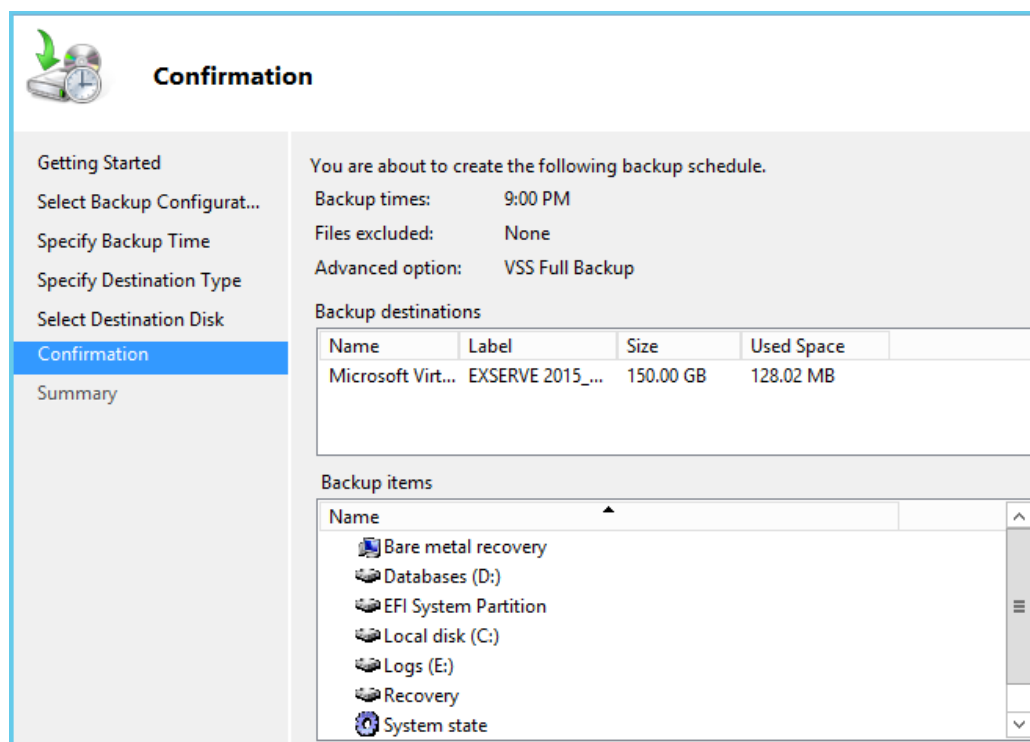
Available disks:

| Disk | Name | Size | Used Space | Volumes in D... |
|---------------------------------------|---------------|-----------|------------|-----------------|
| <input checked="" type="checkbox"/> 3 | Microsoft ... | 150.00 GB | 128.02 MB | |

Show All Available Disks...

At the confirmation screen verify that all your selections are correct.

Note that **VSS Full Backup** is automatically chosen as the backup type, which is correct for this scenario.



Click **Finish** to create the backup job. It will run at the next scheduled time, or if you select **Backup once** from the Actions pane you can run a backup now using the settings that you just configured.

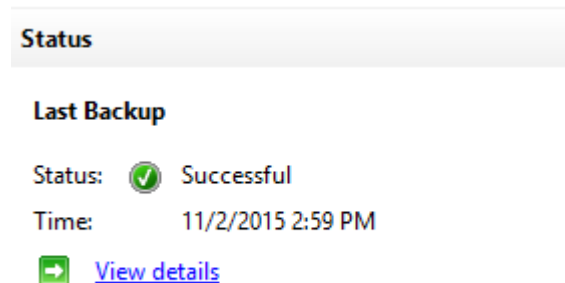
Restoring Mailbox Databases

When an Exchange Server 2016 database has failed you may need to restore it from backup. As an example of this let's look at a scenario where the volume on the server that hosts the database file has been lost due to a hardware failure. The server has been running backups using Windows Server Backup, so we'll restore the lost database from the last successful backup.

In this situation we need to consider what will happen to any new or changed items in mailboxes that has been created or changed since the last backup ran. Obviously the backup itself will not contain those recent changes, however the transaction logs for the database that are stored on a separate volume are still intact. So in this case the logs can be used to roll forward the restored database up to the point in time at which the failure occurred, which

should mean no data loss. However, if the transaction logs were not available, for example if they were on the same volume as the database when it was lost, or because circular logging was enabled, then we would only be able to recover to the point in time at which the backup ran. That would mean accepting some data loss.

My Exchange 2016 backups have been running thanks to a scheduled job in Windows Server Backup. The last backup was successful, so that will be the one used for this recovery.

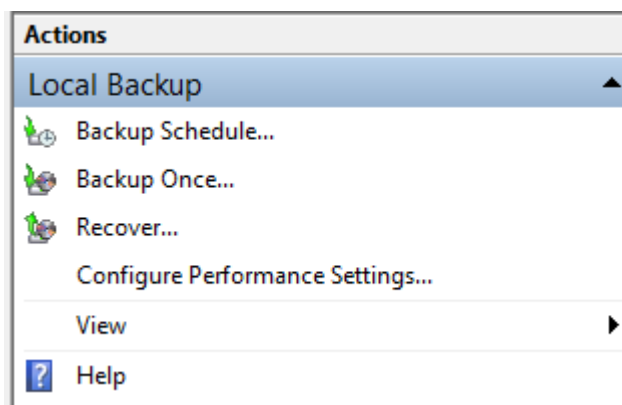


Because I don't trust backup software to tell me the truth I also check the backup time stamp on the database itself.


```
[PS] C:\>Get-MailboxDatabase -Server EX2016SRV1 -Status | fl Name,LastFullBackup  
Name           : DB05  
LastFullBackup : 11/2/2015 2:59:37 PM
```

The failed storage volume has been replaced, formatted and mounted in the same path as it was before. We can now begin the restore.

In Windows Server Backup select **Recover** from the Actions pane.



Select the source of the backup that will be used for recovery. In this example the backup is stored on a volume attached to the server.



Getting Started

Getting Started
Select Backup Date
Select Recovery Type
Select Items to Recover
Specify Recovery Options
Confirmation
Recovery Progress

You can use this wizard to recover files, applications, volumes, or the system state from a backup that was created earlier.

Where is the backup stored that you want to use for the recovery?

☒ This server (EX2016SRV1)

☐ A backup stored on another location

To continue, click Next.

Select the backup that you want to restore from.

Oldest available backup: 11/2/2015 2:59 PM
Newest available backup: 11/2/2015 2:59 PM

Available backups
Select the date of a backup to use for recovery. Backups are available for dates shown in bold.

November 2015

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|----------|-----|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | | | | | |

Backup date: 11/2/2015


Time: 2:59 PM

Location: EX2016S 2015_11_02 ...

Status: Available online

Recoverable items: [Database \(E:\), Logs...](#)

Choose **Applications** as the recovery type.



Select Recovery Type

Select Recovery Type
Getting Started
Select Backup Date
Select Application
Specify Recovery Options
Confirmation
Recovery Progress

What do you want to recover?

☐ Files and folders
You can browse volumes included in this backup and select files and folders.

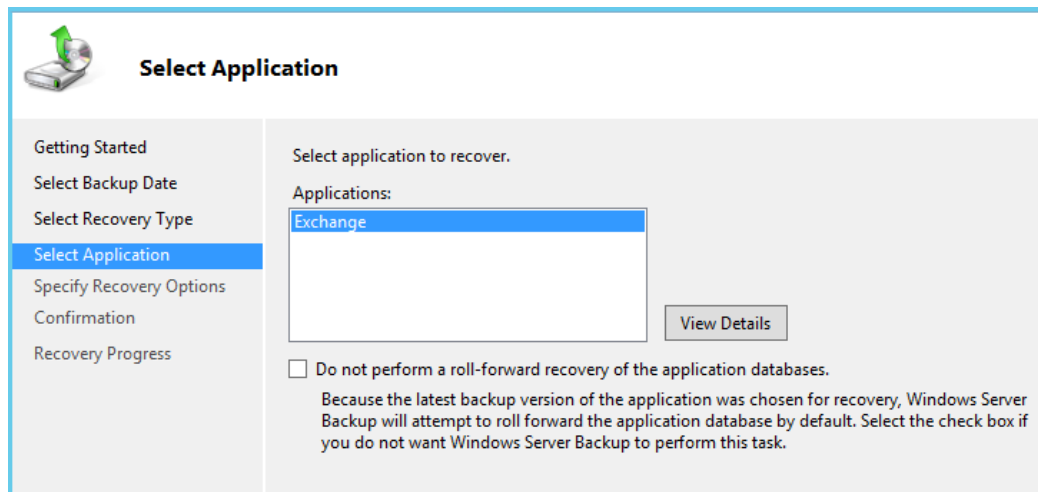
☐ Hyper-V
You can restore virtual machines to their original location, another location or copy the virtual hard disk files of a virtual machine.

☐ Volumes
You can restore an entire volume, such as all data stored on C:.

☒ Applications
You can recover applications that have registered with Windows Server Backup.

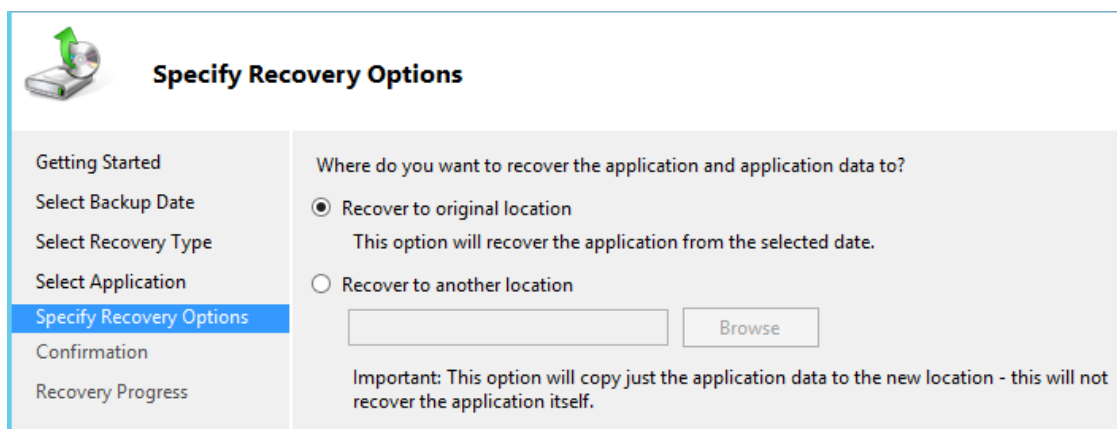
☐ System state
You can restore just the system state.

Select the Exchange application. Note also that there is an option for controlling the roll-forward behaviour. In this example scenario we do want to roll forward the transaction logs and bring the database completely up to date, but if your recovery scenario involves recovering to a specific point in time, or if you have further restores to perform (such as incremental or differential backup sets) then you can check this box to prevent the roll-forward from occurring.



The screenshot shows the 'Select Application' step of the Windows Server Backup recovery wizard. On the left is a navigation pane with the following steps: Getting Started, Select Backup Date, Select Recovery Type, **Select Application** (highlighted), Specify Recovery Options, Confirmation, and Recovery Progress. The main area is titled 'Select Application' and contains the instruction 'Select application to recover.' Below this is a list box labeled 'Applications:' with 'Exchange' selected. To the right of the list box is a 'View Details' button. At the bottom, there is a checkbox labeled 'Do not perform a roll-forward recovery of the application databases.' with the following text: 'Because the latest backup version of the application was chosen for recovery, Windows Server Backup will attempt to roll forward the application database by default. Select the check box if you do not want Windows Server Backup to perform this task.'

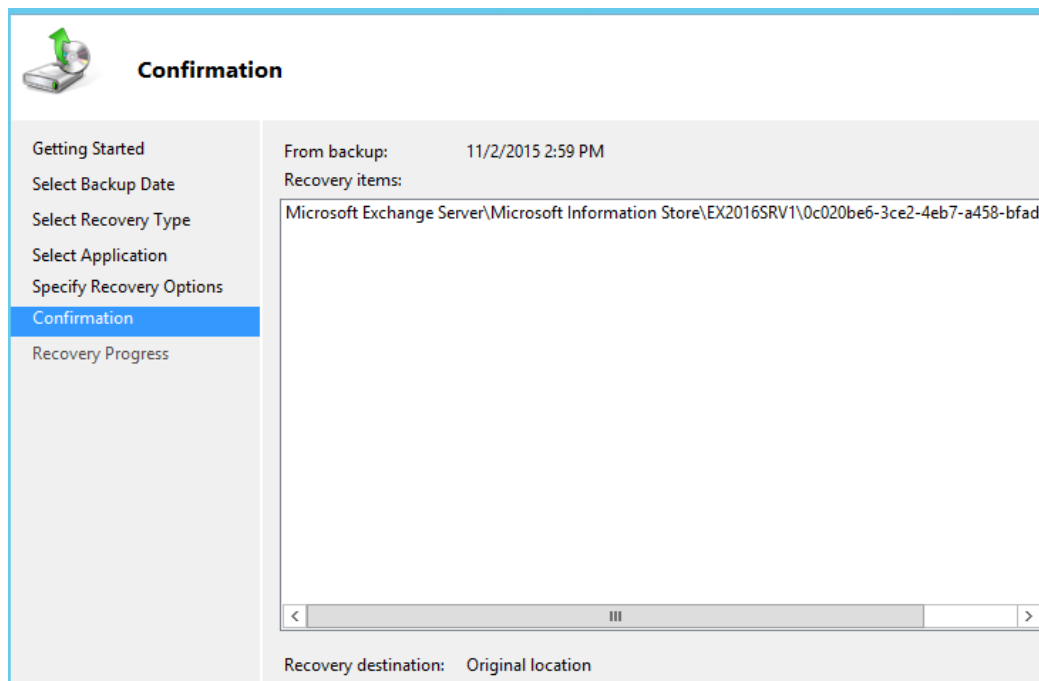
Because we are recovering a completely failed database we want to simply restore to the original location.



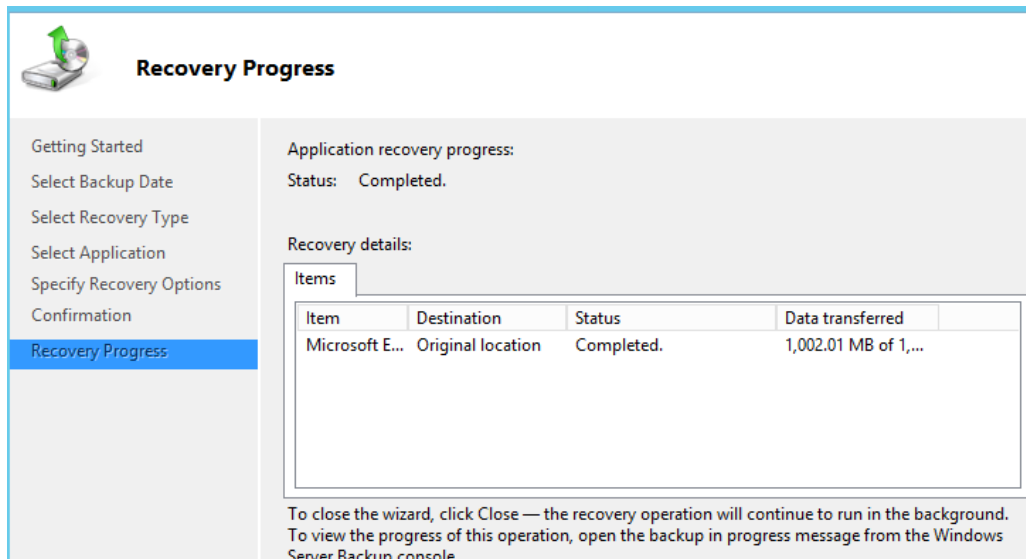
The screenshot shows the 'Specify Recovery Options' step of the Windows Server Backup recovery wizard. On the left is a navigation pane with the following steps: Getting Started, Select Backup Date, Select Recovery Type, Select Application, **Specify Recovery Options** (highlighted), Confirmation, and Recovery Progress. The main area is titled 'Specify Recovery Options' and contains the question 'Where do you want to recover the application and application data to?'. There are two radio button options: 'Recover to original location' (which is selected) and 'Recover to another location'. Below the 'Recover to original location' option is the text 'This option will recover the application from the selected date.' Below the 'Recover to another location' option is a text box and a 'Browse' button. At the bottom, there is a note: 'Important: This option will copy just the application data to the new location - this will not recover the application itself.'

The final step is to confirm the previous selections before beginning the recovery operation.

Warning: When restoring from Windows Server Backup the *entire volume* is restored, not just a specific database. If you are trying to restore just a single database to a volume that has other healthy databases running on it, those databases will be dismounted and included in the restore process as well. This means an outage for the healthy databases, but should not result in data loss if their transaction log files are available for roll-forward to occur. If you are dealing with a live production server and you're unsure about your recovery scenario you can always raise a support ticket with Microsoft Support to assist you with your database recovery.



Click **Recover** to begin the restore. Monitor the restore operation until it completes. If the restore is successful, the database should be mounted automatically for you.



```
[PS] C:\>Get-MailboxDatabase DB05 -Status | select mounted
Mounted : True
```

As you can see a database restore for Exchange Server 2016 is a simple operation when Windows Server Backup is used for backups. Windows Server Backup is supported for production use, and is ideal for test labs where you just want to practice various recovery scenarios. If you use a different backup product to protect your Exchange server then the process will vary, and you should consult the documentation provided by your backup vendor.

Restoring Mailboxes and Items Using a Recovery Database

In Exchange Server 2016 a recovery database allows us to mount a copy of a regular mailbox database on an Exchange server for performing a restore of a mailbox or mailbox items. Recovery databases can only be accessed by administrators performing mailbox restores, they are not accessible by end users via Outlook or any other client application or device, and you will not be able to create new mailboxes on the recovery database.

Using the recovery database will not have an impact on the the active copy of the mailbox database that is running, and a recovery database can be used to mount an Exchange Server 2016 mailbox database from any other server in the same organization.

To recover a mailbox or mailbox items for Exchange Server 2016 using a recovery database the following procedure is used:

- Create the recovery database on an Exchange 2016 server
- Restore a database backup into the recovery database
- Run one or more mailbox restore requests
- Remove the recovery database when it is no longer required

Let's go through the complete process now.

Creating a Recovery Database

To begin with we create the recovery database object itself. Before you do this you should ensure that you have some storage available on the server to host the database and transaction log files that you will be restoring into the recovery database. The storage can be a volume that already hosts other databases, there is no particular need to use a dedicated volume for this purpose unless you need to keep it separate to ensure that the recovery database is excluded from your regular backups. However, I recommend you at least place the recovery database and log files in their own folders away from the active database files.

In the Exchange Management Shell use the `New-MailboxDatabase` cmdlet to create the recovery database. If you know the EDB file name of the database you'll be restoring use it now, but if you don't know it or you need to change it later that's okay.

```
[PS] C:\>New-MailboxDatabase -Server EX2016SRV1 -Name RecoveryDB -Recovery -EdbFilePath  
E:\RecoveryDB\DB05.edb -LogFolderPath F:\RecoveryDB
```

| Name | Server | Recovery | ReplicationType |
|------------|------------|----------|-----------------|
| ---- | ----- | ----- | ----- |
| RecoveryDB | EX2016SRV1 | True | None |

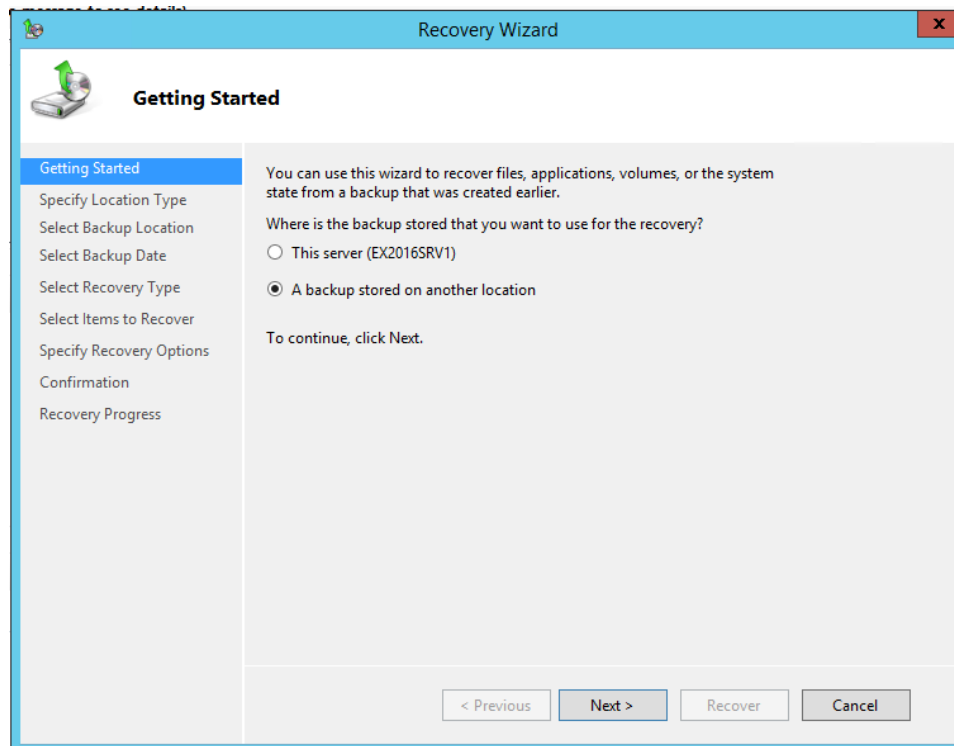
If you inspect the folder paths that you chose for the recovery database you'll notice they are empty. Do not mount the recovery database yet. First we need to restore the database and transaction log files from backup.

Restoring a Mailbox Database into a Recovery Database

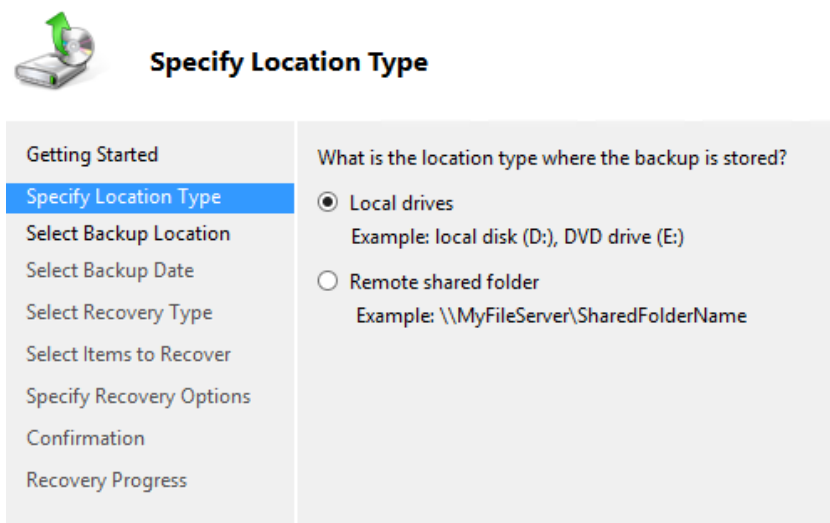
The restoration process for Exchange 2016 databases will depend on the backup software that you're using. You should consult the documentation from your vendor and follow their

guidance. Some backup solutions do not use recovery databases at all, and will use their own utility to open the restored database file and extract the contents that you want to recovery. However, many others will use the recovery database approach, including Windows Server Backup which I am demonstrating here.


In Windows Server Backup click on **Recover** to start a new recovery. I am choosing to recover a backup stored on another location.



The backup drive is attached to this server, so I choose **Local drives** for the location type.



Select the backup drive from the drop down list.



Select Backup Location

Getting Started
Specify Location Type
Select Backup Location
Select Backup Date
Select Recovery Type
Select Items to Recover
Specify Recovery Options
Confirmation
Recovery Progress


Choose the volume or drive that contains the backup. An external disk attached to this server is listed as a volume. If your backup is on a DVD and spans multiple DVDs, make sure that the last DVD of the backup is inserted into a DVD drive.

Backup location:

Total space in location: 99.86 GB

Free space in backup location: 94.57 GB

Select the name of the server from the sets on the backup drive.




Select Server

Getting Started
Specify Location Type
Select Backup Location
Select Server
Select Backup Date
Select Recovery Type
Select Items to Recover
Specify Recovery Options
Confirmation
Recovery Progress

Please select which server's data you would like to recover.

Server:

Select the date from which you want to restore the database and log files.



Select Backup Date

Getting Started
Specify Location Type
Select Backup Location
Select Server
Select Backup Date
Select Recovery Type
Select Items to Recover
Specify Recovery Options
Confirmation
Recovery Progress

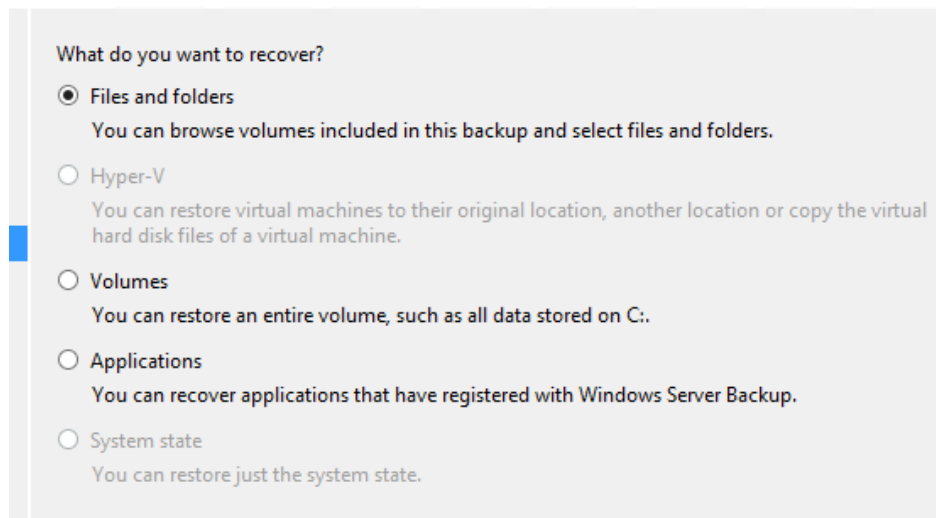
Oldest available backup: 11/2/2015 2:59 PM
Newest available backup: 11/9/2015 1:15 PM

Available backups
Select the date of a backup to use for recovery. Backups are available for dates shown in bold.

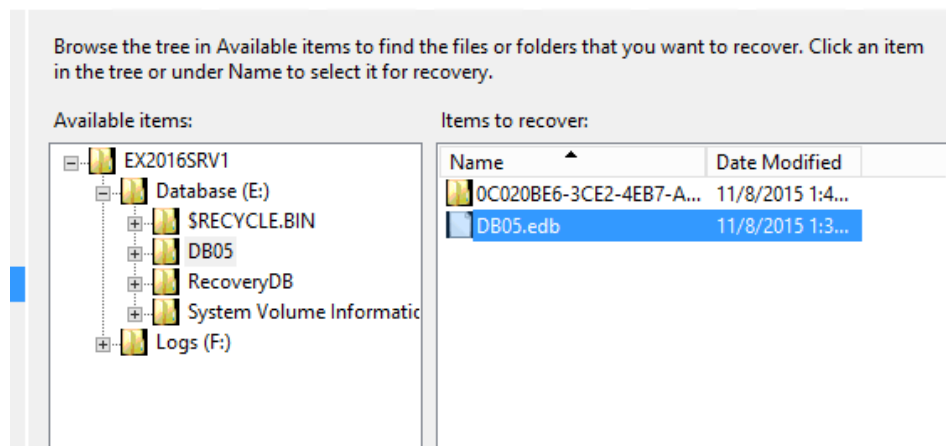
| November 2015 | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|
| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | | | | | |

Backup date: 11/9/2015
Time: 1:15 PM
Recoverable items: [Database \(E:\)](#), [Logs...](#)

In a regular Exchange 2016 database recovery we would choose Applications here, but when working with recovery databases we select **Files and Folders** instead.



From the tree view of available files to restore select the database and log files for the database you're recovering. Note that when the database and logs are in separate folders you'll need to run two separate restore jobs to get them both.



Restore the files to the locations you specified when you created the recovery database earlier.

Recovery destination


☐ Original location
☒ Another location

When this wizard finds items in the backup that are already in the recovery destination

☒ Create copies so that you have both versions
☐ Overwrite the existing versions with the recovered versions
☐ Do not recover the items that already exist on the recovery destination

Security settings

☒ Restore access control list (ACL) permissions to the file or folder being recovered

 File recovery to a non-NTFS target volume might fail due to unsupported file properties.

Confirm your selections and click **Recover** to begin the restore.

From backup: 11/9/2015 1:15 PM

Recovery items:

E:\DB05\DB05.edb

Recovery destination: E:\RecoveryDB

Recovery option: Create copies of recovered files

Security settings: Recover

Repeat the restore for the log files as well if necessary, then move on to making the recovered database mountable.

Making a Restored Database Mountable

If the EDB file you restored is different to the EDB file name you specified when creating the recovery database you'll need to fix that first. To test whether your recovery database is pointing at the correct file use Test-Path:

```
[PS] C:\>Test-Path (Get-MailboxDatabase RecoveryDB).EdbFilePath
True
```

If you do not see a result of “True” use Move-DatabasePath with the **-ConfigurationOnly** switch to update the **EdbFilePath** attribute to the correct path.

```
[PS] C:\>Move-DatabasePath RecoveryDB -EdbFilePath E:\RecoveryDB\DB05.edb -ConfigurationOnly

Confirm
This operation will skip the safety check and make the change to Active Directory directly. Do
you want to continue?
[Y] Yes [A] Yes to All [N] No [L] No to All [?] Help (default is "Y"): y

Confirm
Are you sure you want to perform this action?
Moving database path "RecoveryDB".
```

Next, change directories to the location of the EDB file and use ESEUtil to check the state of the database, which should be in a “Dirty Shutdown” state at the moment. I’ve removed some of the output below for the sake of clarity.

```
[PS] E:\>cd .\RecoveryDB

[PS] E:\RecoveryDB>eseutil /mh .\DB05.edb

Extensible Storage Engine Utilities for Microsoft(R) Exchange Server
Version 15.01
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating FILE DUMP mode...
    Database: .\DB05.edb

Fields:
    State: Dirty Shutdown
```

A soft recovery needs to be performed to get the database file to a state of “Clean Shutdown” so that it will mount. We’ll use ESEUtil and we need to know three things to provide as arguments:

- The database file is located in E:\RecoveryDB (make sure you verify this by looking in Windows Explorer)
- The transaction log files are located in F:\RecoveryDB (again, make sure you verify this)

- The log file prefix is E00 (simply look at the files to see what your prefix is)

That makes the syntax for my soft recovery as follows, running it from the location where the log files are stored:

```
[PS] E:\RecoveryDB>cd F:\RecoveryDB

[PS] F:\RecoveryDB>eseutil /r E00 /l F:\RecoveryDB /d E:\RecoveryDB

Extensible Storage Engine Utilities for Microsoft(R) Exchange Server
Version 15.01
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating RECOVERY mode...
  Logfile base name: E00
    Log files: F:\RecoveryDB
    System files:
  Database Directory: E:\RecoveryDB

Performing soft recovery...
          Restore Status (% complete)

    0    10    20    30    40    50    60    70    80    90   100
  |----|----|----|----|----|----|----|----|----|----|
  .....

Operation completed successfully in 3.265 seconds.
```

Those command-line parameters are:

- /r for “Recovery mode”
- E00 is the log file prefix
- /l (that’s a lower-case L) for the log file path
- /d for the folder path containing the database file (do not include the name of the EDB file itself, only the folder path)

Running ESEUtil /mh again should return a “Clean Shutdown” state this time.

```
[PS] F:\RecoveryDB>cd E:\RecoveryDB
[PS] E:\RecoveryDB>
[PS] E:\RecoveryDB>eseutil /mh .\DB05.edb

Extensible Storage Engine Utilities for Microsoft(R) Exchange Server
Version 15.01
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating FILE DUMP mode...
  Database: .\DB05.edb
Fields:
  State: Clean Shutdown
```

The database is now ready to mount for mailbox recovery.

```
[PS] E:\RecoveryDB>Mount-Database RecoveryDB
```

Running Mailbox Restore Requests

After restoring and mounting the recovery database we can perform one or more mailbox restore requests to retrieve mailbox or mailbox item data. To see which mailboxes are stored within the recovery database use Get-MailboxStatistics.

```
[PS] C:\>Get-MailboxStatistics -Database RecoveryDB | ft -auto
```

| DisplayName | ItemCount |
|---|-----------|
| StorageLimitStatus LastLogonTime | |
| ----- | ----- |
| SystemMailbox{0c020be6-3ce2-4eb7-a458-bfadd3171e86} | 10 |
| HealthMailbox-EX2016SRV1-Mailbox-Database-1808423380 | 5269 |
| ...015 11:45:25 PM | |
| HealthMailbox-EX2016SRV1-001 | 4 |
| ...015 11:48:32 AM | |
| HealthMailbox-EX2016SRV1-006 | 3 |
| In-Place Archive - HealthMailbox-EX2016SRV1-006 | 2 |
| In-Place Archive - HealthMailbox-EX2016SRV1-001 | 2 |
| ...015 12:02:55 PM | |
| In-Place Archive - HealthMailbox-EX2016SRV1-Mailbox-Database-1808423380 | 2 |
| Alannah Shaw | 3710 |
| ...2015 3:57:34 PM | |
| Personal Archive - Alannah Shaw | 3452 |
| ...2015 3:55:54 PM | |
| HealthMailbox-EX2016SRV1-DB05 | 987 |
| ...2015 1:13:51 PM | |
| In-Place Archive - HealthMailbox-EX2016SRV1-DB05 | 2 |

The mailbox for Alannah Shaw is available in this database. Let's look at some different restore scenarios.

To restore a mailbox from the recovery database into the original mailbox we can run New-MailboxRestoreRequest with the following parameters:

```
[PS] C:\>New-MailboxRestoreRequest -Name "Alannah Shaw Recovery" -SourceDatabase RecoveryDB -  
SourceStoreMailbox "Alannah Shaw" -TargetMailbox "Alannah Shaw"
```

If you need to restore to a different folder add the **-TargetRootFolder** parameter as well.

```
[PS] C:\>New-MailboxRestoreRequest -Name "Alannah Shaw Recovery" -SourceDatabase RecoveryDB -  
SourceStoreMailbox "Alannah Shaw" -TargetMailbox "Alannah Shaw" -TargetRootFolder "Mailbox  
Restore"
```

If you need to direct the restored data into the person's archive mailbox instead you simply add the **-TargetIsArchive** switch to the command.

```
[PS] C:\>New-MailboxRestoreRequest -Name "Alannah Shaw" -SourceDatabase RecoveryDB -
SourceStoreMailbox "Alannah Shaw" -TargetMailbox "Alannah Shaw" -TargetIsArchive
```

If you are restoring the mailbox into a different user's mailbox (and this includes if you have *recreated* the user's mailbox to restore into) you will need to add the -

AllowLegacyDNMismatch switch to the command.

```
[PS] C:\>New-MailboxRestoreRequest -Name "Alannah Shaw to Alan Reid" -SourceDatabase RecoveryDB
-SourceStoreMailbox "Alannah Shaw" -TargetMailbox "Alan Reid" -TargetRootFolder "Alannah Shaw
Restore" -AllowLegacyDNMismatch
```

In many restore scenarios you only want to recovery specific folders from the original database. The **-IncludeFolders** parameter can be used to achieve this. There are two ways to nominate the folder names.

For a personal folder (one that the user themselves created) simply use the folder name, and optionally you can include all sub-folders.

```
[PS] C:\>New-MailboxRestoreRequest -Name "Alannah Shaw Invoices Recovery" -SourceDatabase
RecoveryDB -SourceStoreMailbox "Alannah Shaw" -TargetMailbox "Alannah Shaw" -IncludeFolders
"Invoices"

[PS] C:\>New-MailboxRestoreRequest -Name "Alannah Shaw Invoices and Subfolders Recovery" -
SourceDatabase RecoveryDB -SourceStoreMailbox "Alannah Shaw" -TargetMailbox "Alannah Shaw" -
IncludeFolders "Invoices/*"
```

For well-known folders (such as Sent Items, Inbox, Junk Email) the syntax is slightly different.

```
[PS] C:\>New-MailboxRestoreRequest -Name "Alannah Shaw Inbox Recovery" -SourceDatabase
RecoveryDB -SourceStoreMailbox "Alannah Shaw" -TargetMailbox "Alannah Shaw" -IncludeFolders
"#Inbox#"

[PS] C:\>New-MailboxRestoreRequest -Name "Alannah Shaw Inbox and Subfolders Recovery" -
SourceDatabase RecoveryDB -SourceStoreMailbox "Alannah Shaw" -TargetMailbox "Alannah Shaw" -
IncludeFolders "#Inbox#/*"
```

You can also exclude folders from the restore, for example the Sent Items or Deleted Items. The syntax is the same as above and uses the **-ExcludeFolders** parameter.

```
[PS] C:\>New-MailboxRestoreRequest -Name "Alannah Shaw Recovery Excluding Sent Items" -
SourceDatabase RecoveryDB -SourceStoreMailbox "Alannah Shaw" -TargetMailbox "Alannah Shaw" -
ExcludeFolders "#SentItems#", "#DeletedItems#"
```

If you're restoring into a mailbox or folder that already contains items, there may be some conflicts. The **-ConflictResolutionOption** parameter lets you choose one of three options:

- KeepSourceItem (the source in this case is the recovery database's copy of the item, and this is the default option)

- KeepLatestItem (the item with the most recent modified date)
- KeepAll (allows duplicates to be created)

```
[PS] C:\>New-MailboxRestoreRequest -Name "Alannah Shaw" -SourceDatabase RecoveryDB -
SourceStoreMailbox "Alannah Shaw" -TargetMailbox "Alannah Shaw" -ConflictResolutionOption
KeepAll
```

The scenarios above are examples and do not need to be used exclusively. You can combine different features to suit your particular restore scenarios, such as:

- Restoring from Person A's mailbox
- Restoring to Person B's mailbox archive mailbox
- Restoring to a folder named "Person A Restore"
- Excluding Deleted Items and Junk Mail
- Allowing duplicates

All of those can be achieved together in a single mailbox restore request.

Managing Mailbox Restore Requests

You can monitor the progress and any errors for a mailbox restore request by running `Get-MailboxRestoreRequest` and `Get-MailboxRestoreRequestStatistics`.

```
[PS] C:\>Get-MailboxRestoreRequest

Name                               TargetMailbox
-----
Alannah Shaw                     exchangeserverpro.net/Company/Head Office/U...
Completed
Alannah Shaw to Alan Reid        exchangeserverpro.net/Company/Head Office/U...
Completed
Alannah Shaw Recovery Excluding Sent Items exchangeserverpro.net/Company/Head Office/U...
Completed
```

```
[PS] C:\>Get-MailboxRestoreRequest -Name "Alannah Shaw"

Name                               TargetMailbox
-----
Alannah Shaw                     exchangeserverpro.net/Company/Head Office/U...
Completed

[PS] C:\>Get-MailboxRestoreRequest -Name "Alannah Shaw" | Get-MailboxRestoreRequestStatistics
```

| Name | StatusDetail | TargetAlias |
|-----------------|--------------|--------------|
| PercentComplete | | |
| ---- | ----- | ----- |
| ----- | | |
| Alannah Shaw | Completed | Alannah.Shaw |
| 100 | | |

```
[PS] C:\>Get-MailboxRestoreRequest -Name "Alannah Shaw" | Get-MailboxRestoreRequestStatistics |
Select Status*,Include*,
Exclude*,*Transfer*,OverallDuration,CompletedRequestAgeLimit
```

```
Status                : Completed
StatusDetail          : Completed
IncludeFolders         : {}
ExcludeFolders         : {}
ExcludeDumpster        : False
EstimatedTransferSize  : 108.6 MB (113,860,874 bytes)
EstimatedTransferItemCount : 3721
BytesTransferred       : 41.02 MB (43,016,213 bytes)
BytesTransferredPerMinute : 0 B (0 bytes)
ItemsTransferred       : 1557
OverallDuration        : 00:02:57.1693947
CompletedRequestAgeLimit : 3650.00:00:00
```

When completed the mailbox restore request will remain until it is manually removed or until the request age limit has expired. The default age limit is 3650 days (10 years) so you can expect requests to stay around for a long time.

You can specify your own age limit for mailbox restore requests when you first create them by adding the **-CompletedRequestAgeLimit** parameter and providing a value such as "7.00:00:00" which is 7 days.

Otherwise, to remove the completed restore requests use the **Remove-MailboxRestoreRequest** cmdlet.

```
[PS] C:\>Get-MailboxRestoreRequest | Where Status -eq Completed | Remove-MailboxRestoreRequest
```

Removing a Recovery Database

When you are finished with your data recovery you can remove the recovery database from the Exchange 2016 server. First, dismount the database.

```
[PS] C:\>Dismount-Database RecoveryDB

Confirm
Are you sure you want to perform this action?
Dismounting database "RecoveryDB". This may result in reduced availability for mailboxes in the
database.
[Y] Yes [A] Yes to All [N] No [L] No to All [?] Help (default is "Y"): y
```

Then remove the database.

```
[PS] C:\>Remove-MailboxDatabase RecoveryDB

Confirm
Are you sure you want to perform this action?
Removing mailbox database "RecoveryDB".
[Y] Yes [A] Yes to All [N] No [L] No to All [?] Help (default is "Y"): y
WARNING: The specified database has been removed. You must remove the database file located in
E:\RecoveryDB\DB05.edb
from your computer manually if it exists. Specified database: RecoveryDB
```

Finally, delete the files from the file system on your server to reclaim that disk space.

Recovering a Failed Exchange 2016 Server

At some point in your career as a hard-working IT professional you're likely to encounter a scenario in which an Exchange 2016 server has completely failed. For example, you might have a physical server with a dead hardware component preventing it from booting, or a virtual machine with a corrupt operating system volume that is preventing it from starting up.

There are many ways out of these types of situations, and I won't dictate that one particular solution is the best one in all cases. You'll have a variety of factors to take into account in your particular scenario, for example perhaps the hardware component that failed can be quickly replaced and the server brought online, negating the need for any other software-driven recovery process to take place.

But let's assume for the sake of this demonstration that your situation involves a failed server that will not boot, and you either have replacement server hardware that you can use or are planning to spin up another virtual machine to run Exchange 2016.

In either case a recovery installation of Exchange Server 2016 may be the appropriate course of action for you to take. At a high level the process goes like this:

1. You install a new Windows Server instance with the same characteristics as the failed server (the same server name, Windows Server version, drive letters, and performance/capacity)
2. You perform a recovery install of Exchange 2016 by running setup with the **/mode:recoverserver** switch

3. You re-apply any custom configurations that were not automatically re-applied by the recovery install
4. You restore the Exchange databases if those volumes were also lost in the server failure.

Let's take a look at an example of a recovery installation for Exchange Server 2016.

Preparing the Server for Recovery

To begin you'll need to prepare the server that you'll be recovering Exchange Server 2016 on. Whether that is the same server or a different server depends on your specific circumstances, but either way you'll need to:

- Install the same Windows Server operating system and service pack level on the server
- Configure your storage volumes to use the same drive letters as the previous server
- Join the server to the domain (note that you will first need to reset the computer account that already exists in Active Directory)
- Install the Exchange Server 2016 pre-requisites

Performing a Recovery Install of Exchange 2016

On the server that you've prepared copy the Exchange Server 2016 setup files to a location where they'll be accessible on the server. You should use the same setup files for the Exchange Server 2016 cumulative update that was previously installed on the server, do not try to use this recovery process to upgrade or downgrade the server.

Open an elevated command prompt and change to the folder containing the Exchange 2016 setup files. Run the following command.

```
C:\> setup /mode:recoverserver /IAcceptExchangeServerLicenseTerms
```

Restoring Custom Configurations

There is often confusion around what counts as a "custom configuration" in this scenario. Exchange Server 2016 stores most of its configuration in Active Directory, such as virtual directory URL settings, transport settings, the names and locations of databases, and so on.

However, anything that is machine-specific or stored locally on the server such as IIS settings, SSL certificates, modified configuration files, or registry keys, will not be restored by the

recovery install. You should always document and automate the post-install configuration of your Exchange servers so that any such customizations can be reapplied in this type of scenario.

On the most basic of Exchange 2016 deployments you're likely to have to at least reinstall the SSL certificate for the server. If you have multiple Exchange 2016 or 2013 servers that have the same SSL certificate installed then you can simply export the certificate from one of those servers, and then import it to the recovered server. Note that the steps for this are the same for Exchange 2016 as they are for Exchange 2013. After reinstalling the SSL certificate, you can then also enable it for Exchange services.

Restoring Databases After Server Recovery

If the volumes that contain your mailbox databases and transaction logs were not lost in the server failure, and are configured with the same drive letters or mount points as before, then you will likely find that the databases are able to successfully mount and continue operation without data loss.

If the databases and log files are available, but will not mount, you may need to perform a soft recovery of the database first using ESEUtil.

If the databases and log files were lost in the server failure, then you will need to restore them from your last backup following the procedures demonstrated earlier in this chapter.

What Next?

What you've just read is enough to give you a good understanding of the fundamental concepts of Exchange Server 2016, and how to set up a server to perform the basic tasks of hosting mailboxes and allow users to send and receive emails.

If you've read this far without getting hands on with the software yourself, then I recommend that you use the lab setup guide in the appendix of this eBook to set up a test environment that you can use.

If you have already set up your test environment as you worked through this eBook, then you now have a functioning email server that you can use to further your skills in Exchange Server 2016. Every time you encounter a real world challenge or question you can dive into your test lab, do some research on the internet, and work out a solution without risking any production systems.

To stay up to date with what's happening in the Exchange Server and Office 365 world you can [subscribe to the Exchange Server Pro newsletter here](#).

And if you have any feedback or questions about this eBook, then I'm happy to read them. You can get in touch using [my contact form here](#).

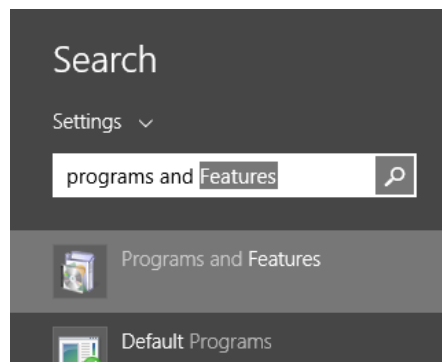
Thanks for reading!

Appendix: Lab Setup Guide

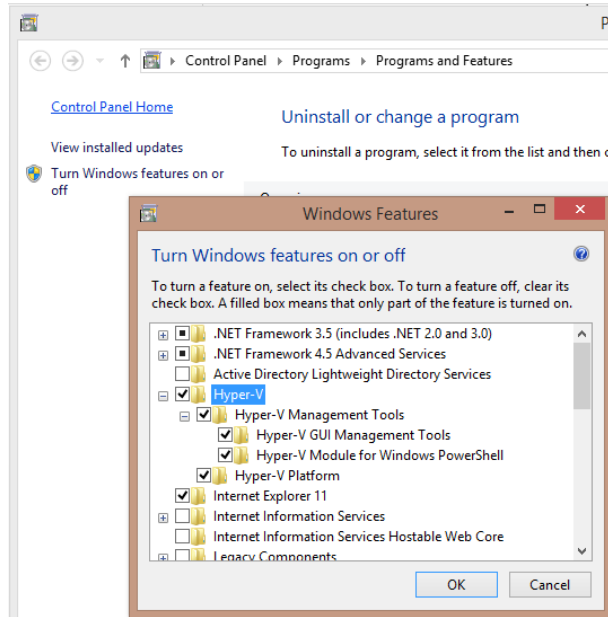
In this appendix we'll go through the preparation steps for setting up a test lab environment using Hyper-V on Windows 8.1 or Windows 10.

Installing Hyper-V on Windows 8.1 or Windows 10

Although Windows 8.1 and Windows 10 ship with Hyper-V included as a feature it is not enabled by default. To enable Hyper-V first open **Programs and Features**. You can access Programs and Features by clicking the Start button and performing a search.



Click on **Turn Windows Features On or Off** and make sure that Hyper-V and all sub-components are ticked.



If Windows prompts you to restart after enabling these then go ahead and reboot your computer. While you are rebooting you should also go into your system BIOS and confirm that virtualization support is enabled. This may appear under one of a variety of names such as “Intel VT” or “Hardware-assisted Virtualization Technology”. Refer to your motherboard manual if you need more guidance.

Downloading Software

To build a test lab environment for Exchange Server 2016 using the latest software available at the time of writing this guide you should download the following from Microsoft.

- Windows Server 2012 R2. This will be an ISO file. You can run Windows Server 2012 R2 as a time-limited trial.
- Exchange Server 2016. Check for the latest Cumulative Update and download just that one.
- Windows 7 with Service Pack 1, or a later version of Windows client. Some people prefer Windows 7 for virtual machines because it has a simpler user interface and will run with relatively low memory allocated.

Newer versions of the software listed above may be available by the time you are reading this, so you should double check whether there is a more recent build available. For example, if Exchange Server 2016 is up to Cumulative Update 1, just download that one instead of the RTM build.

Generally speaking you can just download the latest version available at the time, however there may be minor changes to features or user interfaces that make things slightly different for you than what is demonstrate in this guide.

What Else Do You Need?

In addition to a host machine and the software for installing your virtual machines you should also consider purchasing your own domain name if you do not already own one.

Owning your own domain name means that you can establish real inbound/outbound mail flow between your test lab environment and the rest of the world. You'll also be able to remotely access Exchange services such as Outlook on the web and ActiveSync.

The demonstrations throughout the Exchange Server 2016 Quick Start Guide use the domain name exchange2016demo.com. Obviously you should not try to use the exact same domain name that is used as a demonstration in this guide.

However if you do accidentally create your Active Directory with the same DNS name as ours, or a DNS name that is not valid on the internet (eg "domain.local"), you can still register your own domain name to be used for email addresses in your Exchange Server 2016 organization.

Building Virtual Machines in Hyper-V

To save on disk space you can build your virtual machines by first creating a base image that is then used as the source disk for other virtual machines. This is achieved using a Hyper-V feature called Differencing Disks.

If you prefer to just install each virtual machine independently without using Differencing Disks, but this will use more disk space on your host machine.

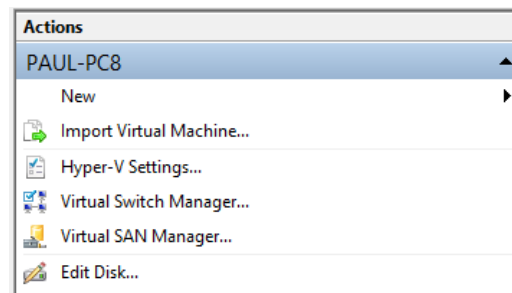
In addition to using Differencing Disks we can also use a dynamically sized disk for the base image, so that a reasonable system volume size can be allocated (eg 120Gb) without actually consuming all of that disk space on your host machine.

We know that 120Gb might sound like a lot of disk space just for a test server, but Exchange Server 2013 expects to see a large system volume so it is better to give it one. Use the combination of Differencing Disks and Dynamically Expanding disks reduce your overall disk space usage.

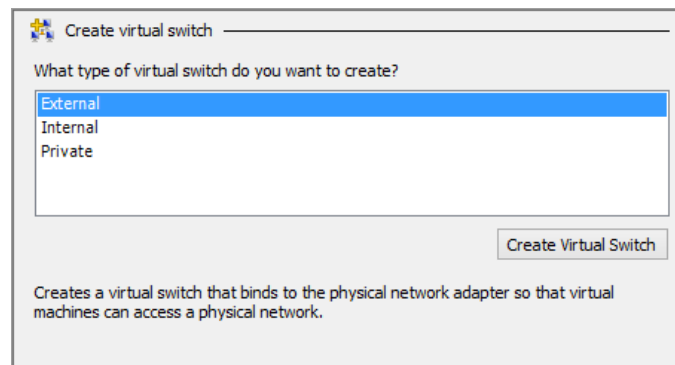
Configuring a Virtual Switch

If this is the first time you've used Hyper-V on your host machine you'll need to configure a virtual switch for the virtual machines to connect to.

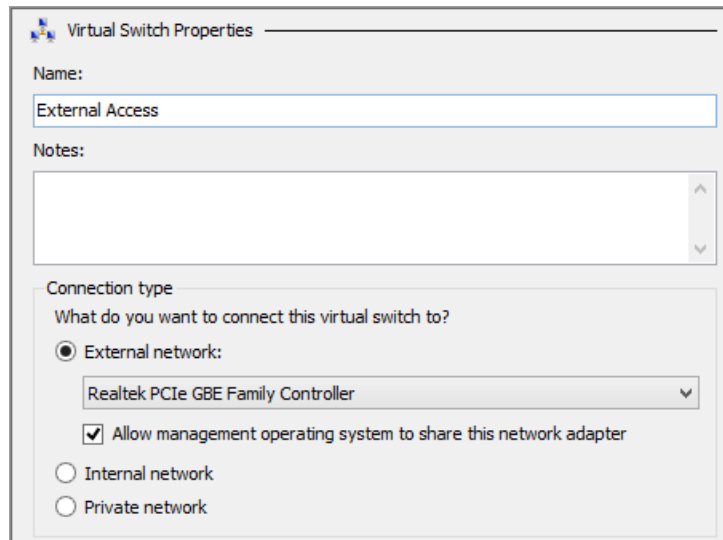
Open Hyper-V Manager and click **Virtual Switch Manager**.



Choose **External** as the type of virtual switch, and click **Create Virtual Switch**.



Give your virtual switch a meaningful name such as "External Access", and configure it to use a network adapter in your host machine that is connected to your local area network. If you only have one network adapter you can share it with the management operating system (your host machine's operating system) as well.

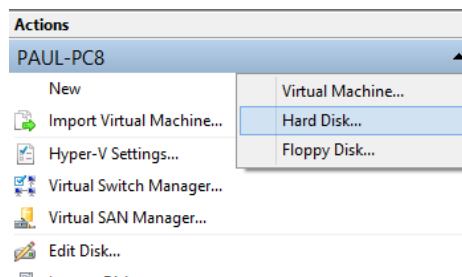


Click **OK** or **Apply** to finish creating the new virtual switch.

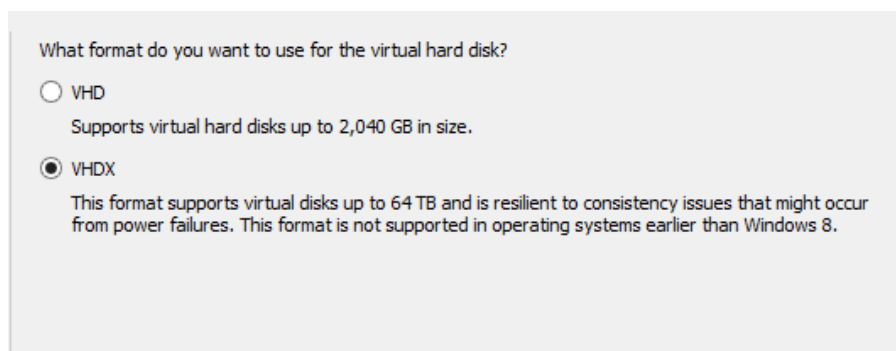
Creating a Virtual Hard Disk to Use as a Base Image

Before you create the new virtual machine itself you should first create the virtual hard disk.

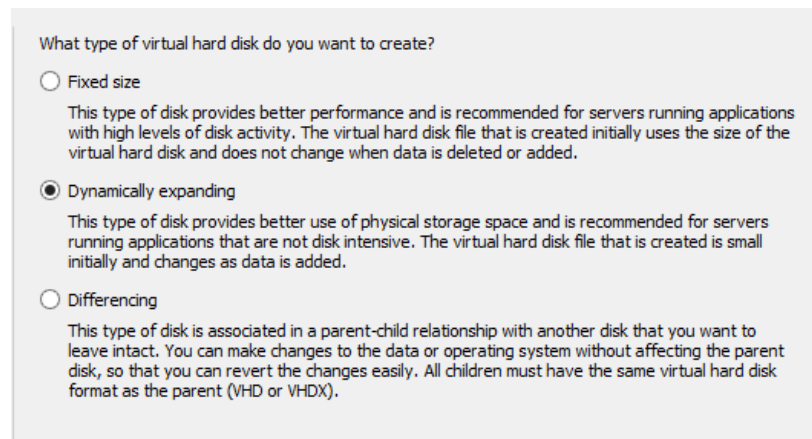
Open Hyper-V Manager and click **New → Hard Disk**.



Skip past the Before You Begin page. Choose the **VHDX** format for your virtual hard disk and click **Next** to continue.



Choose the disk type of **Dynamically Expanding**, and click **Next** to continue.



What type of virtual hard disk do you want to create?

☐ Fixed size

This type of disk provides better performance and is recommended for servers running applications with high levels of disk activity. The virtual hard disk file that is created initially uses the size of the virtual hard disk and does not change when data is deleted or added.

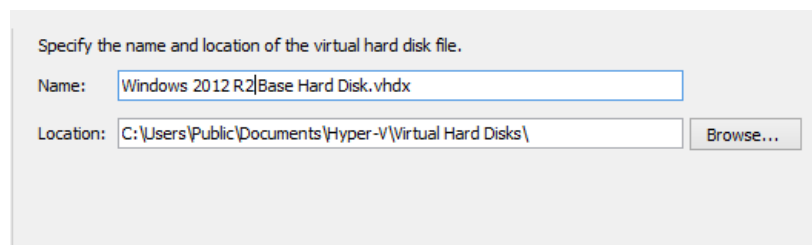
☒ Dynamically expanding

This type of disk provides better use of physical storage space and is recommended for servers running applications that are not disk intensive. The virtual hard disk file that is created is small initially and changes as data is added.

☐ Differencing

This type of disk is associated in a parent-child relationship with another disk that you want to leave intact. You can make changes to the data or operating system without affecting the parent disk, so that you can revert the changes easily. All children must have the same virtual hard disk format as the parent (VHD or VHDX).

Give the virtual hard disk a meaningful name such as “Window 2012 R2 Base Hard Disk” and choose a location for the file to be placed. Click **Next** to continue.

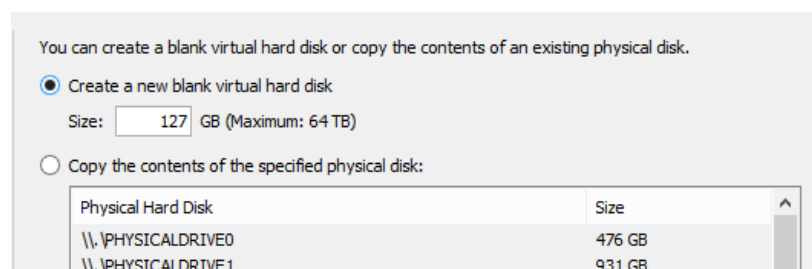


Specify the name and location of the virtual hard disk file.

Name:

Location:

The default size of 127Gb is fine for our usage. Click **Next** to continue.



You can create a blank virtual hard disk or copy the contents of an existing physical disk.

☒ Create a new blank virtual hard disk

Size: GB (Maximum: 64 TB)

☐ Copy the contents of the specified physical disk:

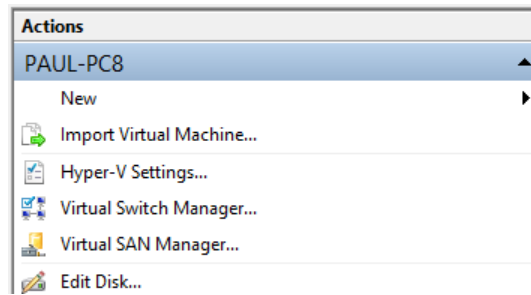
| Physical Hard Disk | Size |
|--------------------|--------|
| \\.\PHYSICALDRIVE0 | 476 GB |
| \\.\PHYSICALDRIVE1 | 931 GB |

Double-check your choices on the summary page and if you’re happy with everything click **Finish** to create the virtual hard disk.

Creating a Virtual Machine to Use as a Base Image

Now that we've configured your virtual switch and a virtual hard disk for your base image we can go ahead and install the virtual machine that will be used as a base image for all of our other Windows Server 2012 R2 virtual machines.

Open Hyper-V Manager and click **New → Virtual Machine** to create a new virtual machine.



Skip past the Before You Begin page, and give your new virtual machine a name. Because this is eventually going to be used as a base image we've called this one "Windows 2012 R2 Base Image". Click **Next** to continue.

Choose a name and location for this virtual machine.


The name is displayed in Hyper-V Manager. We recommend that you use a name that helps you easily identify this virtual machine, such as the name of the guest operating system or workload.

Name:

You can create a folder or use an existing folder to store the virtual machine. If you don't select a folder, the virtual machine is stored in the default folder configured for this server.

☐ Store the virtual machine in a different location

Location:


 If you plan to take checkpoints of this virtual machine, select a location that has enough free space. Checkpoints include virtual machine data and may require a large amount of space.

Although Generation 1 virtual machines will work just fine, we can choose Generation 2 as we are installing a compatible guest operating system. Click **Next** to continue.

Choose the generation of this virtual machine.

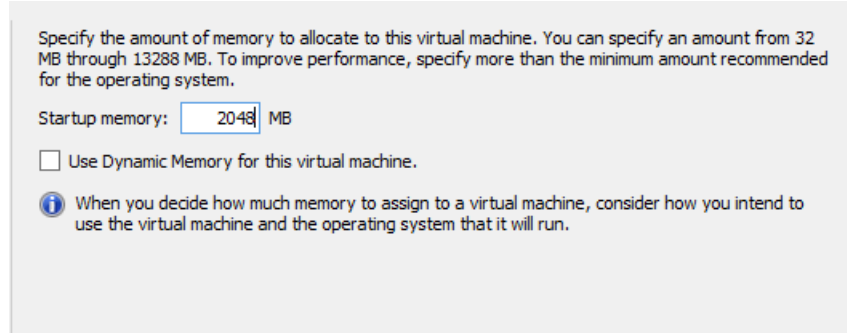
☐ Generation 1
This virtual machine generation provides the same virtual hardware to the virtual machine as in previous versions of Hyper-V.

☒ Generation 2
This virtual machine generation provides support for features such as Secure Boot, SCSI boot, and PXE boot using a standard network adapter. Guest operating systems must be running at least Windows Server 2012 or 64-bit versions of Windows 8.

 Once a virtual machine has been created, you cannot change its generation.

Assign enough memory so that your virtual machine will perform well while you are configuring it in readiness to become your base image.


We've assigned 2Gb in this case. Click **Next** to continue.



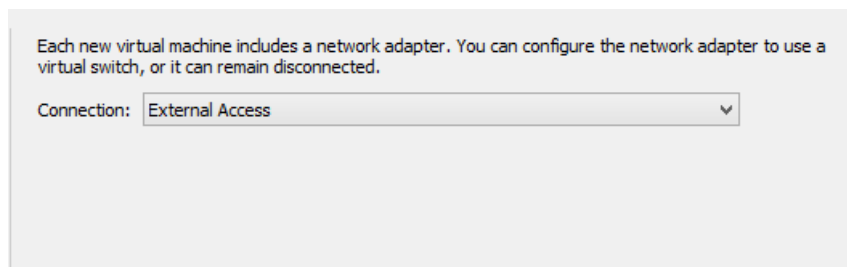
Specify the amount of memory to allocate to this virtual machine. You can specify an amount from 32 MB through 13288 MB. To improve performance, specify more than the minimum amount recommended for the operating system.

Startup memory: MB

☐ Use Dynamic Memory for this virtual machine.

 When you decide how much memory to assign to a virtual machine, consider how you intend to use the virtual machine and the operating system that it will run.

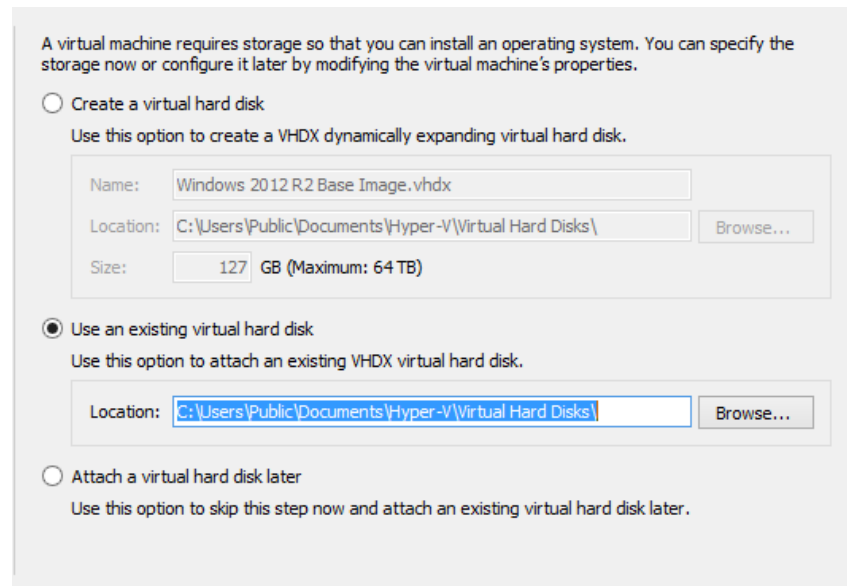
Select the virtual switch that you configured to allow your virtual machines to connect to your network and the internet. Click **Next** to continue.



Each new virtual machine includes a network adapter. You can configure the network adapter to use a virtual switch, or it can remain disconnected.

Connection:

Select **Use an existing virtual hard disk** and browse to the location where your VHDX file was created.



A virtual machine requires storage so that you can install an operating system. You can specify the storage now or configure it later by modifying the virtual machine's properties.

☐ Create a virtual hard disk
Use this option to create a VHDX dynamically expanding virtual hard disk.

Name:

Location:

Size: GB (Maximum: 64 TB)

☒ Use an existing virtual hard disk
Use this option to attach an existing VHDX virtual hard disk.

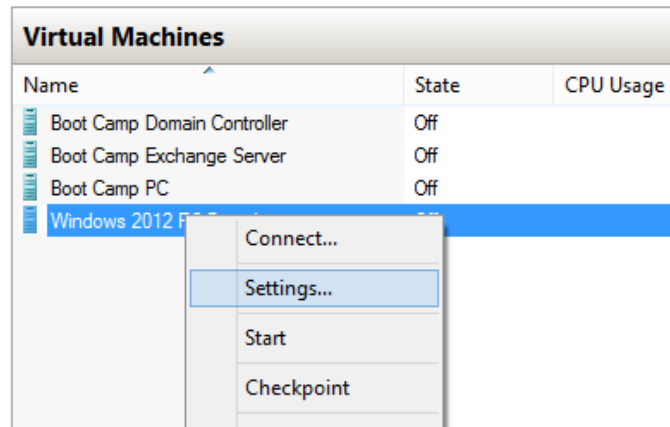
Location:

☐ Attach a virtual hard disk later
Use this option to skip this step now and attach an existing virtual hard disk later.

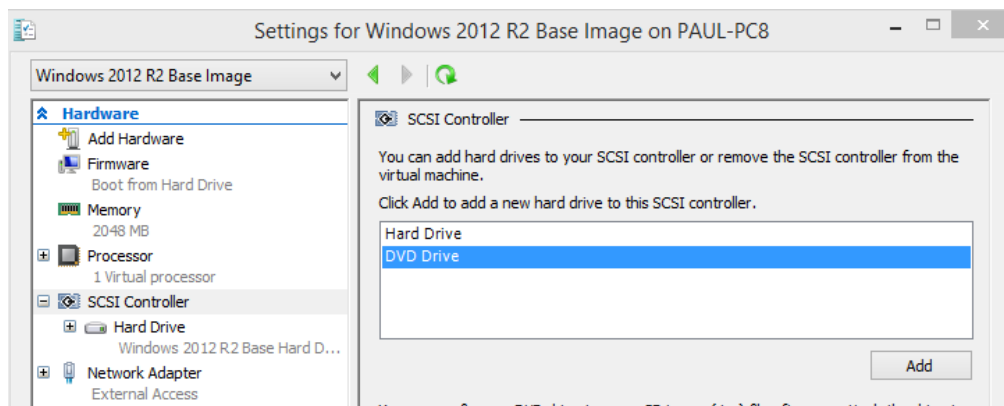
After selecting the file click **Next** to continue.

Review your selections on the summary page and click **Finish** to create the virtual machine.

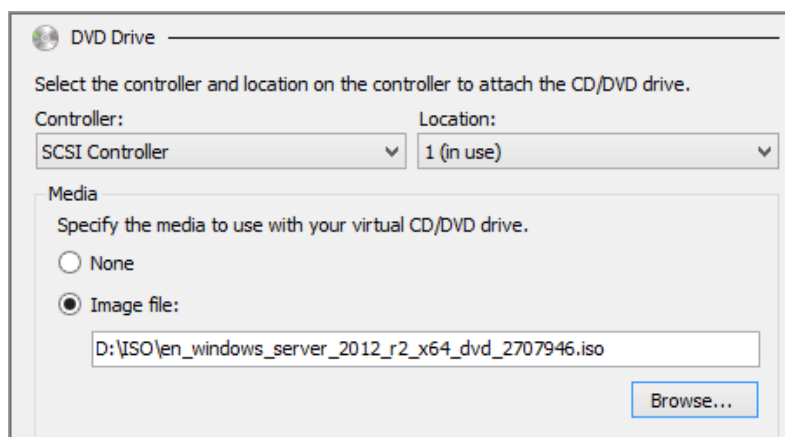
After creating the virtual machine we need to connect the ISO file that contains the Windows Server 2012 R2 installation media. In Hyper-V Manager right-click the virtual machine and choose **Settings**.



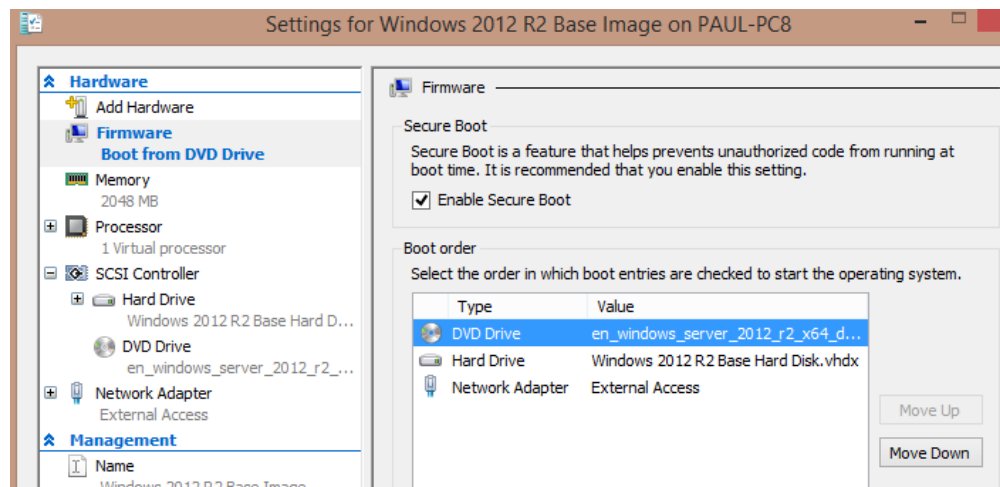
Select the virtual SCSI Controller and add a DVD drive.



Click **Browse** and select the ISO file that you have downloaded for Windows Server 2012 R2.

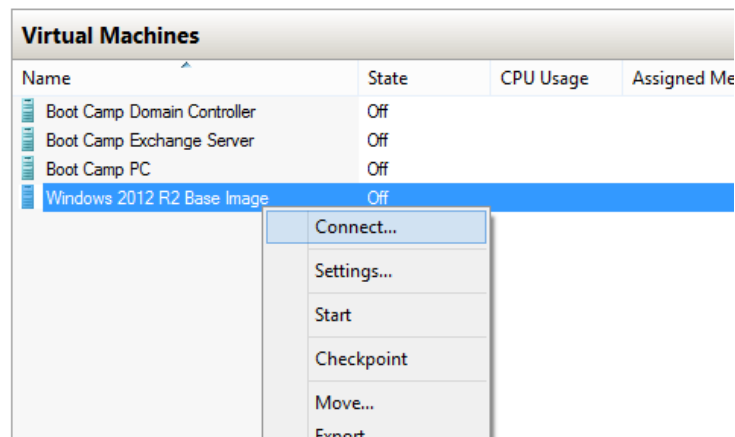


Click **Apply** to apply the hardware changes. Next, select the Firmware settings and move the DVD Drive to the top of the boot order.

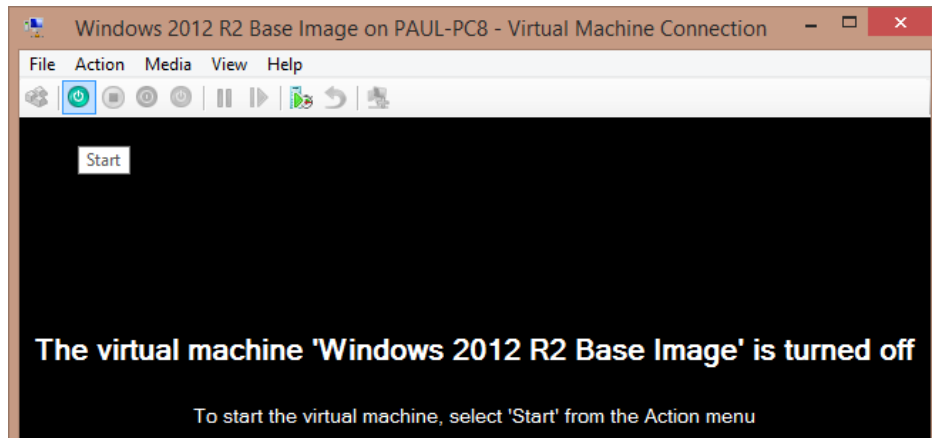


Click **OK** to apply the changes and close the Settings window.

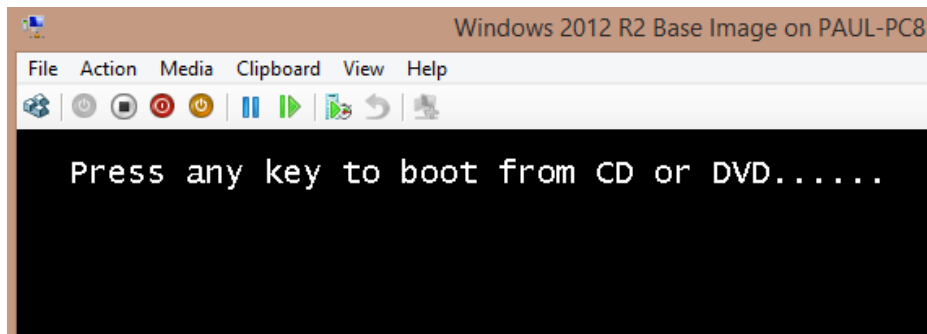
Connect to the virtual machine so that you can see the console.



Click the power icon to start the virtual machine.



At the prompt press any key to boot from the DVD drive.

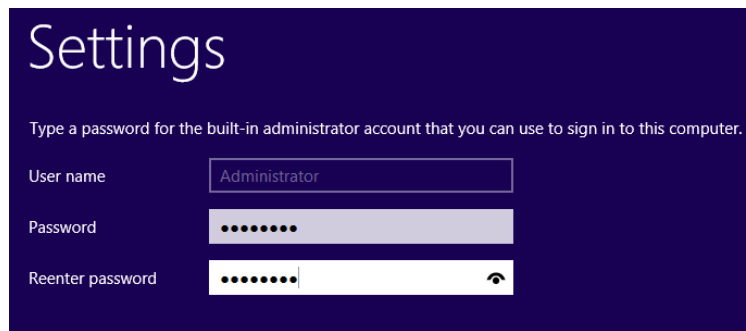


Windows Server 2012 R2 setup will begin.



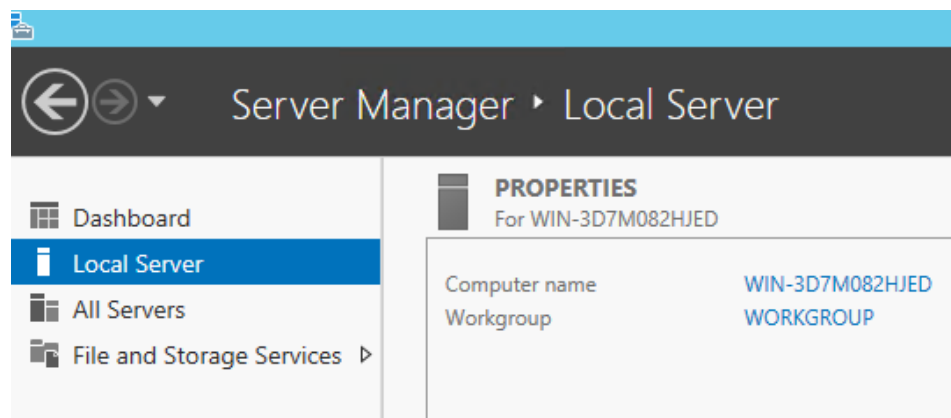
There is very little customization to perform during this stage of setup. Simply follow the setup prompts choosing the appropriate language and regional settings for your country, enter your product key (one is issued to you if you download the trial software), and make sure you choose to install “**Server with a GUI**” not “Server Core Installation”.

On first boot Windows will prompt you to choose an administrator password.



After Windows has finished booting log in using the administrator password you chose.

The Server Manager console will automatically launch. Choose **Local Server** from the left hand side.

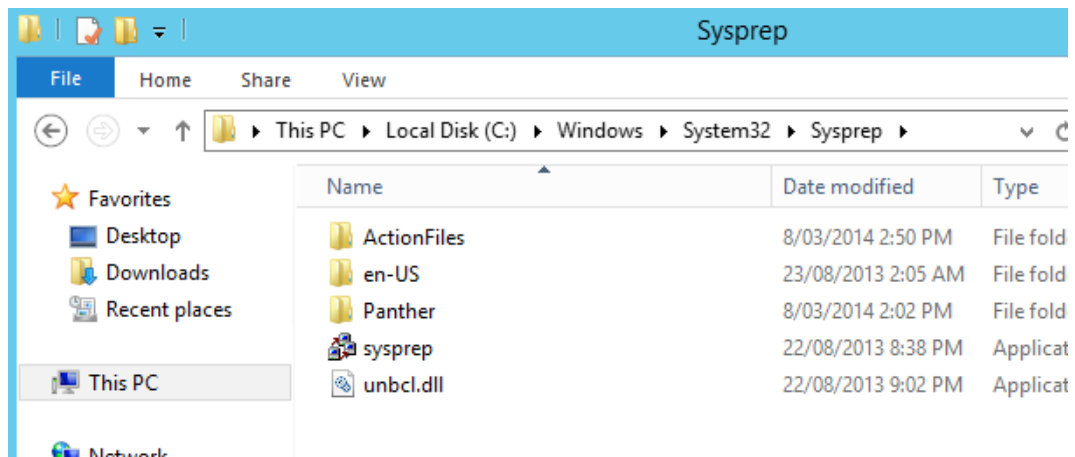


Because this virtual machine will become a base image for other virtual machines there is only a small amount of customization that we want to do at this stage.

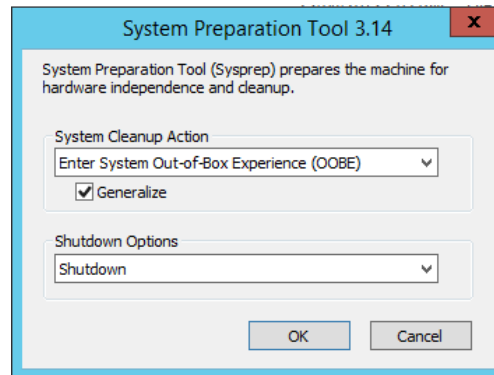
- Configure Windows Update and run an update to install any patches that are available
- Set the time zone to suit your location
- Turn off IE Enhanced Security Configuration
- Enable Remote Desktop

After finishing those initial configurations and restarting for Windows updates to finish installing we can Sysprep the server.

The Sysprep files are located in **C:\Windows\System32\Sysprep**.

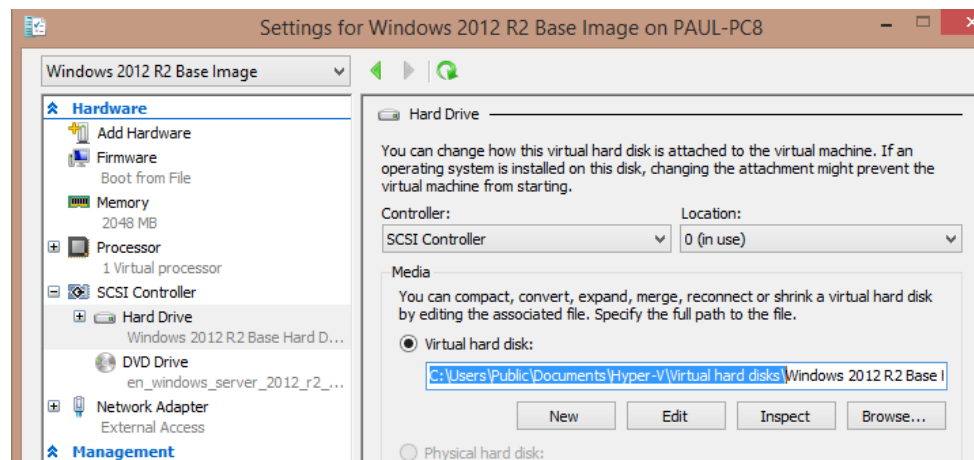


Run **Sysprep** and tick the box to **Generalize**, and set the **Shutdown Options** to “Shutdown”.

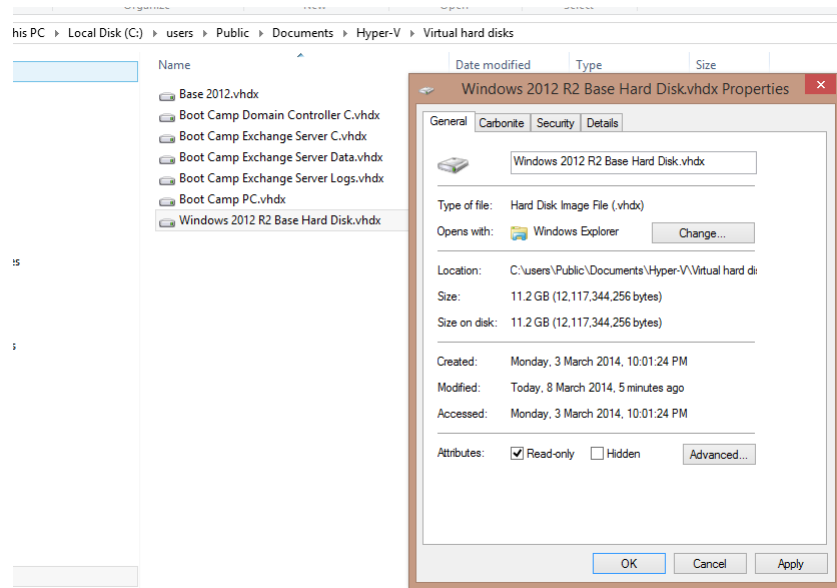


Click **OK** and wait a few minutes for the virtual machine to shut down.

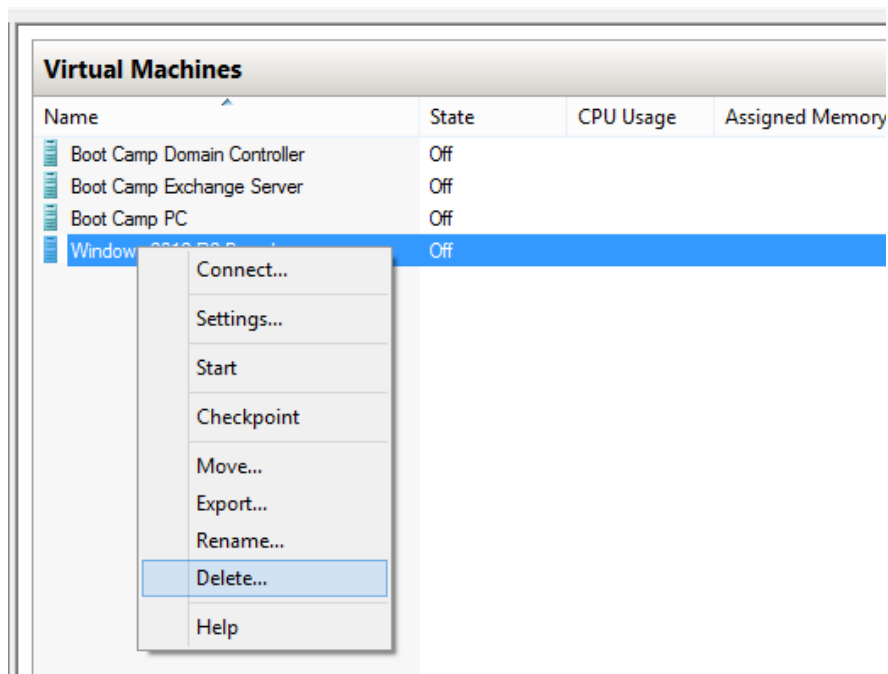
Open the virtual machine settings again and look for the location of the virtual hard drive.



Navigate to the location in Windows Explorer and set the file to **Read Only**.



In Hyper-V Manager we can also delete the virtual machine itself. This will leave the virtual hard drive on your computer to be used as the base image for other virtual machines.

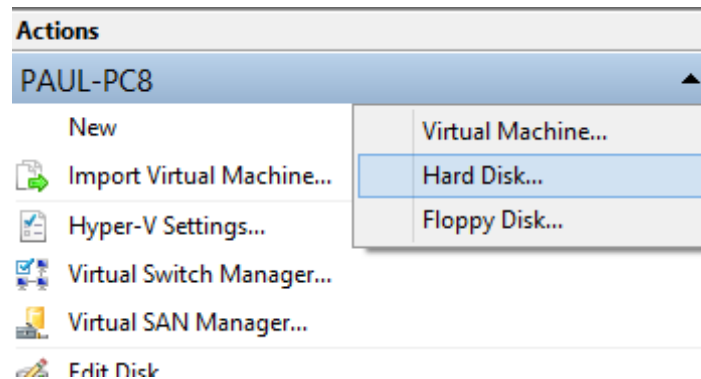


Now we can begin building virtual machines to use in the test lab environment

Installing the Domain Controller

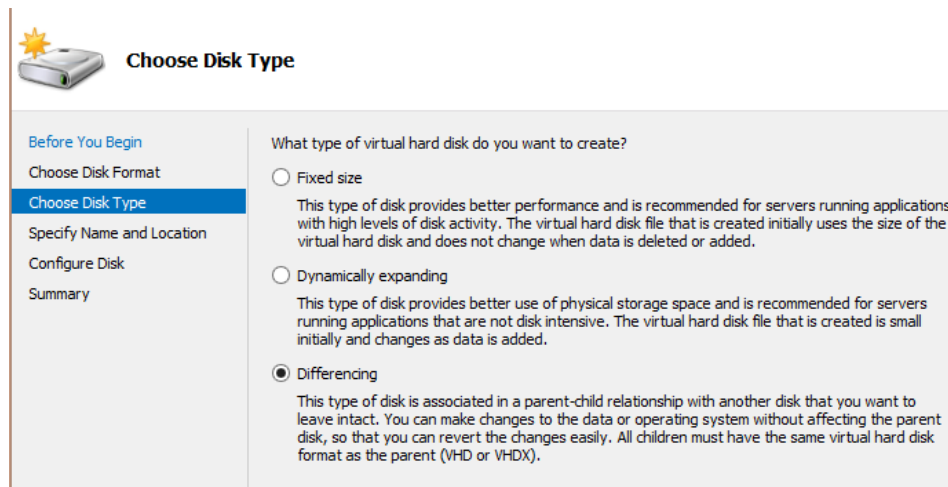
To begin the setup of a virtual machine to be the Active Directory domain controller we first need to create the differencing disk for the virtual machine.

In Hyper-V Manager select **New → Hard Disk**.

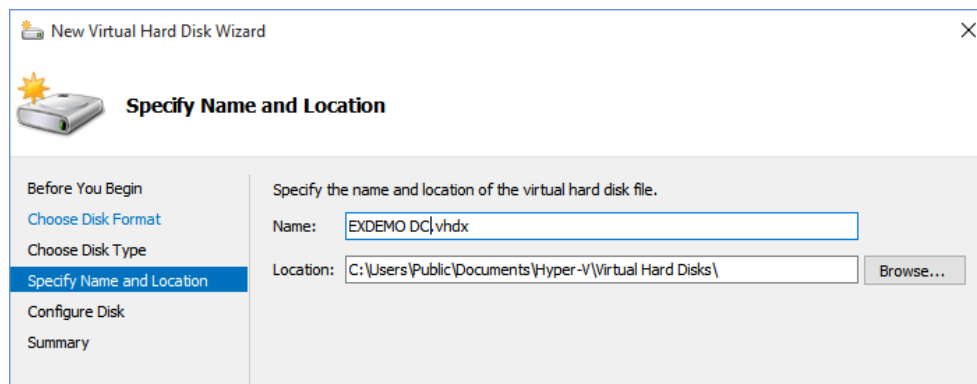


Create a new hard disk in VHDX format.

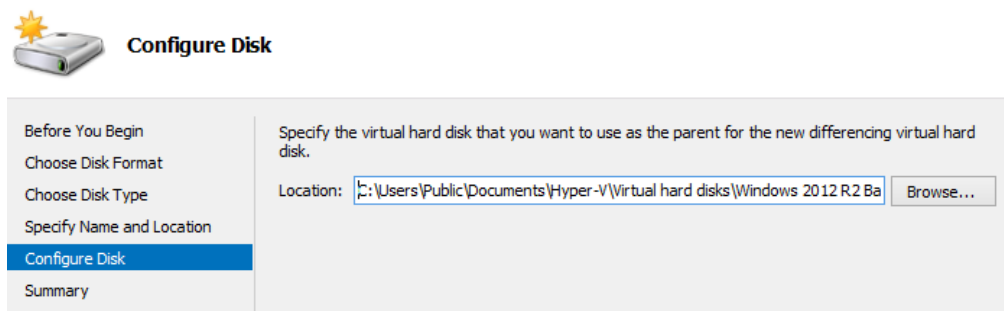
Choose the Disk Type of **Differencing**.



Give the hard disk a meaningful file name and choose a location to store it.

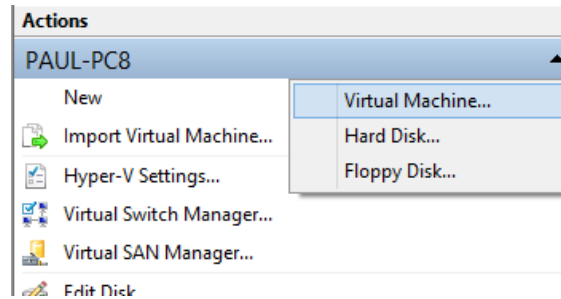


For the parent drive select the virtual hard drive that you created as a base image in the previous section of this chapter.



On the Summary page verify your selections then click **Finish** to create the virtual hard drive.

Next, create a new virtual machine.



Follow the same steps to create the virtual machine as you used in the previous section of this chapter.

- Give the virtual machine a unique, meaningful name such as “EXDEMO DC” or “Test Lab DC”
- Give the virtual machine at least 2Gb of memory if possible
- Connect the virtual machine to your virtual switch
- When connecting a virtual hard disk choose to use an existing virtual hard disk and select the differencing disk you created earlier in this section

Tip: If you have a host machine with 16Gb of memory you might run into situations later where one of your Exchange servers won’t start in Hyper-V due to insufficient available memory. One of the ways you can work around this is to configure your domain controller virtual machine to use Dynamic Memory, and set a lower values for minimum and startup memory amount, such as 512Mb minimum and 1024mb startup, with a maximum memory of 2048Mb.

After creating the virtual machine we can power it on and enter regional settings, a product key, accept the license, and enter a local administrator password.

When we look at Server Manager we can see that the base image configurations for Remote Desktop, IE Enhanced Security Configuration, and the Time Zone are already set the same for the new server.

So we just need to:

- Enable Windows Update again
- Configure a static IP address
- Configure a computer name

The computer name change will require a restart.

After restarting we can log back in and go to Server Manager again.

In the **Manage** menu select **Add Roles and Features**.

Skip past the Before You Begin page and choose **Role-based or feature-based installation**. Click **Next** to continue.

Select the server from the server pool. Click **Next** to continue.

Select both **Active Directory Domain Services** and **DNS Server** from the list of roles. When prompted to add other required features and management tools as well simply click **Add Features** to accept those additions. Click **Next** to continue.

Select server roles

DESTINATION SERVER
DC

Before You Begin

Installation Type

Server Selection

Server Roles

Features

AD DS

DNS Server

Confirmation

Results

Select one or more roles to install on the selected server.

Roles

- ☐ Active Directory Certificate Services
- ☒ Active Directory Domain Services
- ☐ Active Directory Federation Services
- ☐ Active Directory Lightweight Directory Services
- ☐ Active Directory Rights Management Services
- ☐ Application Server
- ☐ DHCP Server
- ☒ DNS Server
- ☐ Fax Server
- ☐ File and Storage Services (2 of 12 installed)

Description

Domain Name System (DNS) Server provides name resolution for TCP/IP networks. DNS Server is easier to manage when it is installed on the same server as Active Directory Domain Services. If you select the Active Directory Domain Services role, you can install and configure DNS Server and Active Directory Domain Services to work together.

Click **Next** again to progress past the Features, AD DS, and DNS Server pages after you have read the information on each one.

On the Confirmation page tick the box to automatically restart if required, so that you don't need to manually restart it, although a restart usually will not be required at this stage anyway.

Confirm installation selections

DESTINATION SERVER
DC

Before You Begin

Installation Type

Server Selection

Server Roles

Features

AD DS

DNS Server

Confirmation

Results

To install the following roles, role services, or features on selected server, click Install.

- ☒ Restart the destination server automatically if required

Optional features (such as administration tools) might be displayed on this page because they have been selected automatically. If you do not want to install these optional features, click Previous to clear their check boxes.

Active Directory Domain Services
DNS Server
Group Policy Management
Remote Server Administration Tools
 Role Administration Tools
 AD DS and AD LDS Tools
 AD DS Tools
 Active Directory Administrative Center
 AD DS Snap-Ins and Command-Line Tools
DNS Server Tools

[Export configuration settings](#)
[Specify an alternate source path](#)

Activate Windows
Go to System in Control P
activate Windows

< Previous

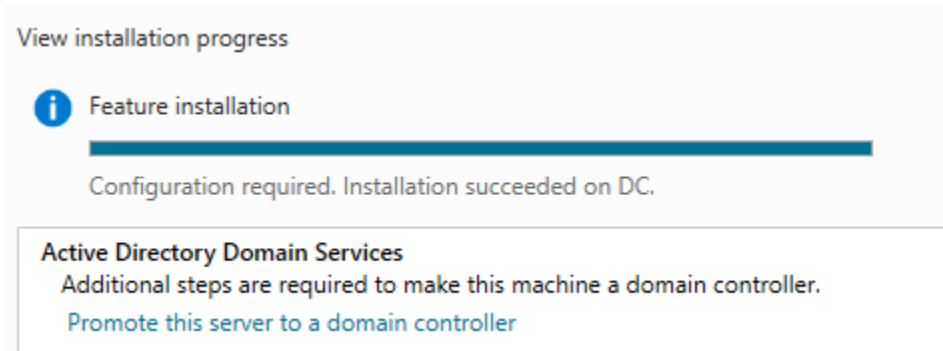
Next >

Install

Cancel

Click **Install** to begin the installation of the server roles.

When the installation has completed click the link to **Promote this server to a domain controller**.



Choose to Add a new forest. Enter your root domain name. This is where you can use your own domain name that you own as your Active Directory namespace. I'm using exchange2016demo.com.

Deployment Configuration TARGET SERVER
DC

Deployment Configuration

Domain Controller Options

Additional Options

Paths

Review Options

Prerequisites Check

Installation

Select the deployment operation

☐ Add a domain controller to an existing domain

☐ Add a new domain to an existing forest

☒ Add a new forest

Specify the domain information for this operation

Root domain name:

Note: If you do not own a domain you can also use a namespace such as "domain.local". Just be aware that using a domain name that you don't genuinely own, may cause you some issues later on (for example we don't recommend you use the same domain name as us). However if you do use a .local or similar name for now, you can still buy a domain name later if you want to establish real inbound/outbound email capabilities for your test lab environment.

Forest and Domain functional levels of Windows Server 2012 R2 are okay to use with Exchange Server 2016. Enter a Directory Services Restore Mode password and click **Next** to continue.

Domain Controller Options

TARGET SERVER
DC

Deployment Configuration

Domain Controller Options

DNS Options

Additional Options

Paths

Review Options

Prerequisites Check

Installation

Results

Select functional level of the new forest and root domain

Forest functional level:Windows Server 2012 R2

Domain functional level:Windows Server 2012 R2

Specify domain controller capabilities

☒ Domain Name System (DNS) server

☒ Global Catalog (GC)

☐ Read only domain controller (RODC)

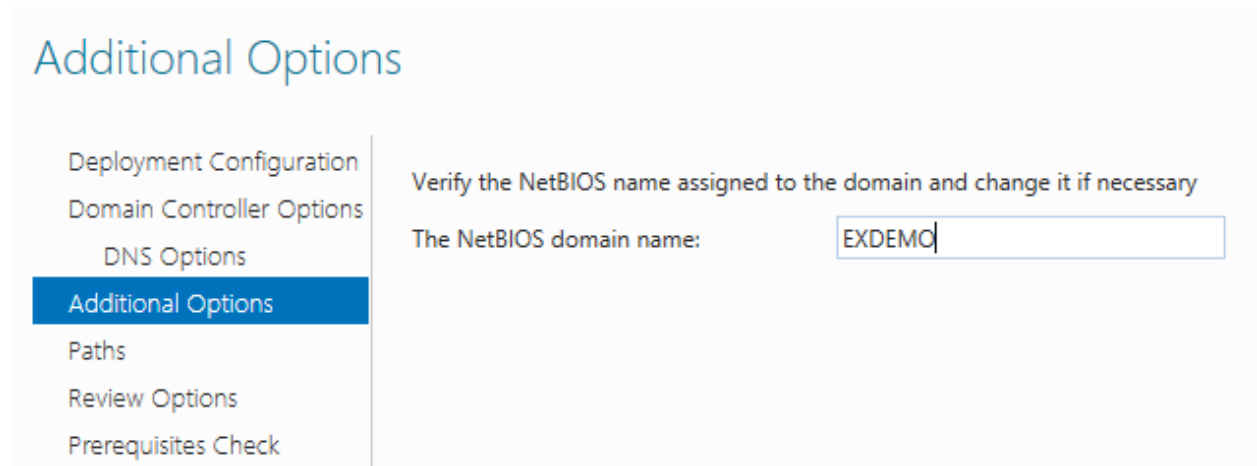
Type the Directory Services Restore Mode (DSRM) password

Password:••••••••

Confirm password:••••••••

Disregard any DNS delegation warnings and click **Next** to continue.

Enter a NetBIOS name for the domain and click **Next** to continue.



The default file paths for the AD DS database, logs and SYSVOL are fine for a test lab environment, so click **Next** to continue.

Review your selections and click **Next** to continue.

After the prerequisites check is complete click **Install**. The server will restart to complete the process.

Tip: To simplify your test lab environment you might like to use Group Policy Management to modify the Default Domain Policy to disable password expiry. However if you plan to open up access to your test lab from the internet (eg, Outlook Web App or ActiveSync) then complex passwords should still be used.

Install Certificate Services

Exchange Server 2013 makes use of SSL certificates to secure client-server connectivity, such as for Outlook Web App, ActiveSync, and Outlook Anywhere.

Although SSL certificates can be purchased relatively cheaply from commercial providers, for a test lab environment you can run your own Certificate Authority (CA) and issue SSL certificates for your Exchange servers at no cost.

A downside to this method is that non-domain members (such as mobile devices) will not trust your CA and will display certificate trust warnings when connecting to your Exchange servers (or may not connect at all).

In Server Manager open the **Manage** menu and select **Add Roles and Features**.

Skip past the Before You Begin page and choose **Role-based or feature-based installation**. Click **Next** to continue.

Select your server from the Server Pool and click **Next** to continue.

Tick the box for **Active Directory Certificate Services** and when prompted for additional required features click **Add Features**.

| Roles | Description |
|--|--|
| <input checked="" type="checkbox"/> Active Directory Certificate Services | Active Directory Certificate Services (AD CS) is used to create certification authorities and related role services that allow you to issue and manage certificates used in a variety of applications. |
| <input checked="" type="checkbox"/> Active Directory Domain Services (Installed) | |
| <input type="checkbox"/> Active Directory Federation Services | |
| <input type="checkbox"/> Active Directory Lightweight Directory Services | |
| <input type="checkbox"/> Active Directory Rights Management Services | |

Click **Next** to continue past the Features and AD CS pages.

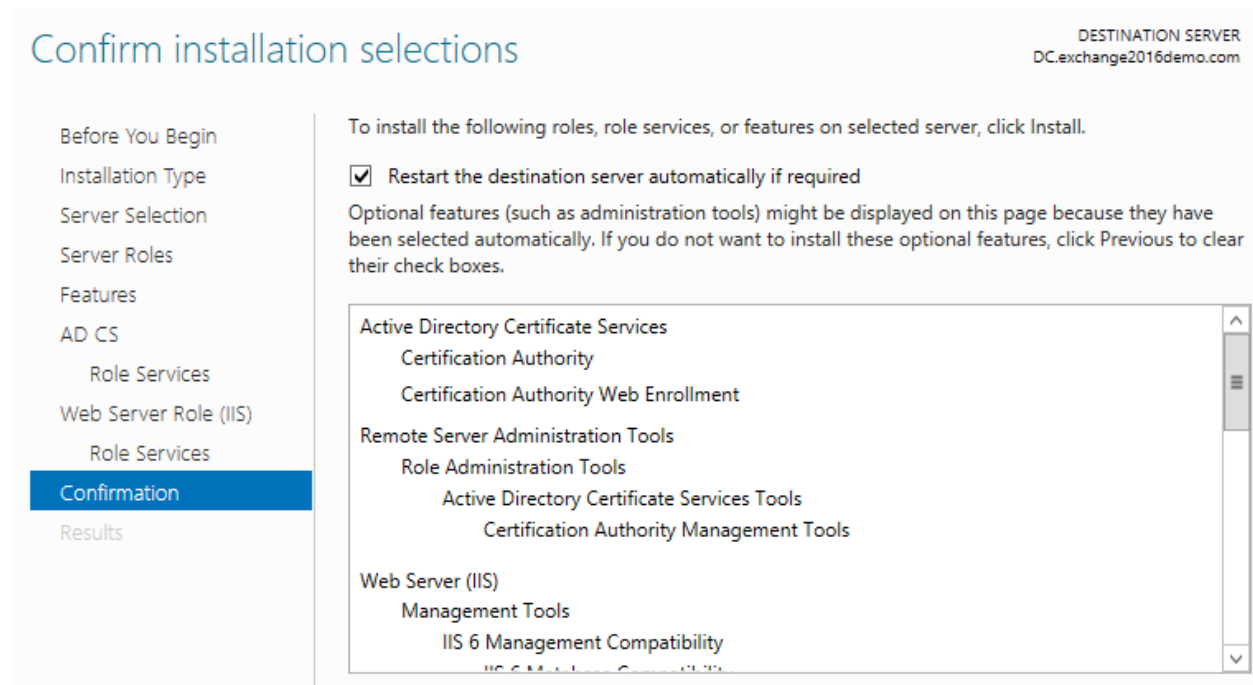
On the Role Services page tick the box for **Certification Authority Web Enrollment**, and click **Add Features** when prompted for additional required features.

Click **Next** to continue.

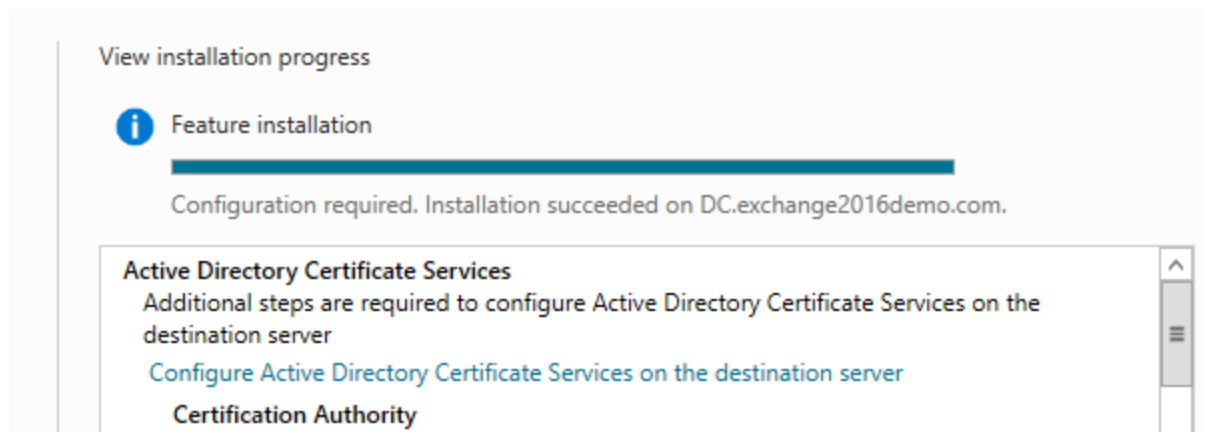
| Role services | Description |
|--|---|
| <input checked="" type="checkbox"/> Certification Authority | |
| <input type="checkbox"/> Certificate Enrollment Policy Web Service | |
| <input type="checkbox"/> Certificate Enrollment Web Service | |
| <input checked="" type="checkbox"/> Certification Authority Web Enrollment | Certification Authority Web Enrollment provides a simple Web interface that allows users to perform tasks such as request and renew certificates, retrieve certificate revocation lists (CRLs), and enroll for smart card certificates. |
| <input type="checkbox"/> Network Device Enrollment Service | |
| <input type="checkbox"/> Online Responder | |

Click **Next** twice more, then on the Confirmation page tick the box to automatically restart if required (although it usually is not required).

Click **Install** to begin the role installation.



After installation has complete click the link to **Configure Active Directory Certificate Services on the destination computer**.



Use your Administrator credentials to configure the role. Click **Next** to continue.

Credentials

DESTINATION SERVER
DC.exchange2016demo.com

Credentials

Role Services

Confirmation

Progress

Results

Specify credentials to configure role services

To install the following role services you must belong to the local Administrators group:

- Standalone certification authority
- Certification Authority Web Enrollment
- Online Responder

To install the following role services you must belong to the Enterprise Admins group:

- Enterprise certification authority
- Certificate Enrollment Policy Web Service
- Certificate Enrollment Web Service
- Network Device Enrollment Service

Credentials: EXDEMO\Administrator

Change...

Tick both role services and click **Next** to continue.

Role Services

DESTINATION SERVER
DC.exchange2016demo.com

Credentials

Role Services

Setup Type

CA Type

Private Key

Cryptography

CA Name

Validity Period

Certificate Database

Confirmation

Select Role Services to configure

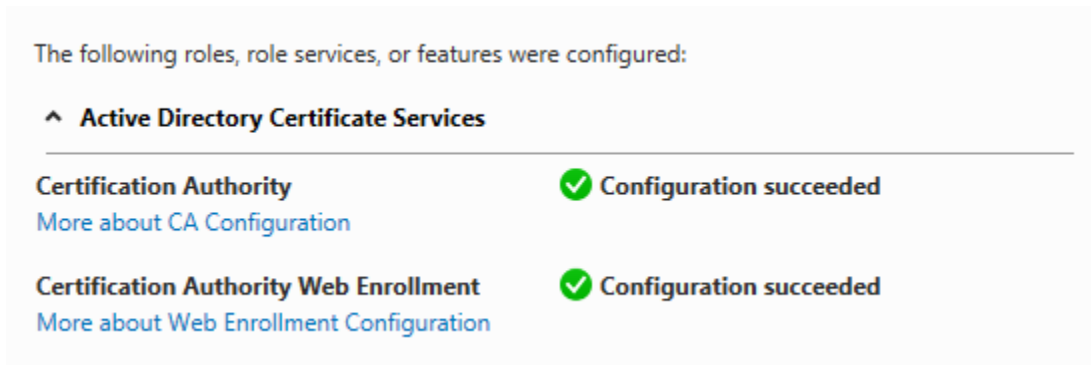
☒ Certification Authority
 ☒ Certification Authority Web Enrollment
 ☐ Online Responder
 ☐ Network Device Enrollment Service
 ☐ Certificate Enrollment Web Service
 ☐ Certificate Enrollment Policy Web Service

Continue through the wizard. Most of the selections you will be making are the default option.

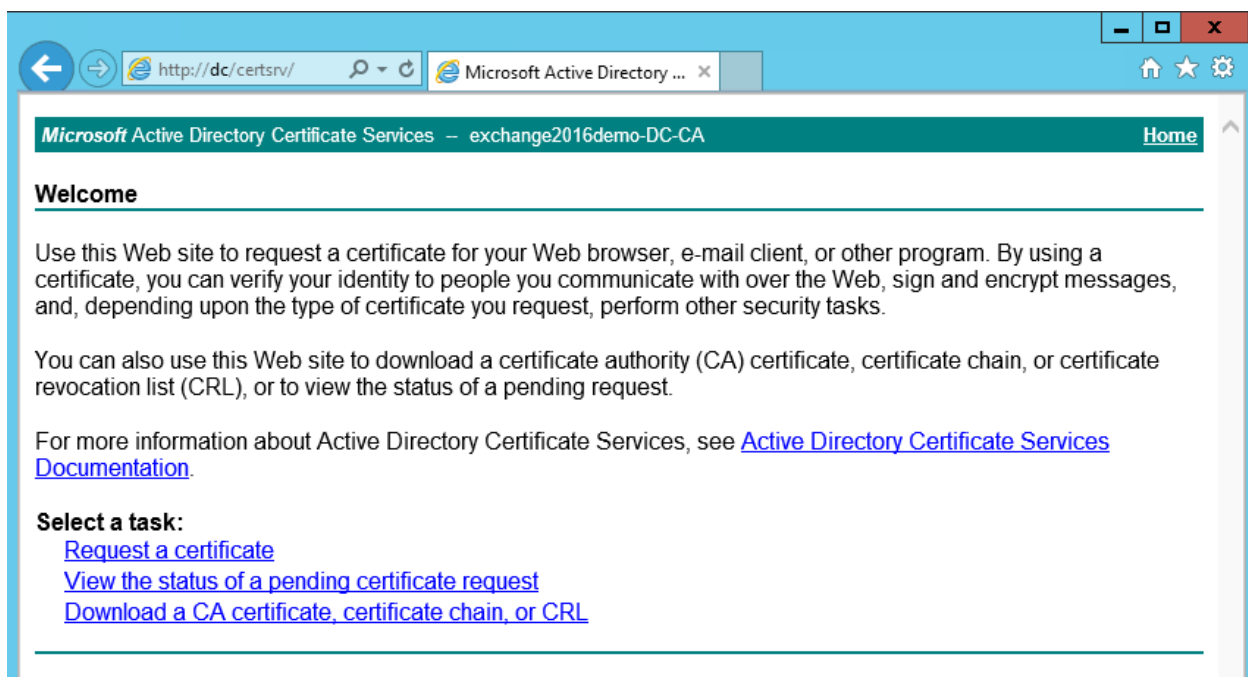
- Choose to configure an **Enterprise CA**
- Specify a CA type of **Root CA**
- Create a **new private key**
- Accept the default **cryptographic provider**
- Accept the default **common name** for the CA (or you can change it if you wish)
- Accept the default **validity period** of 5 years
- Accept the default **database and log locations**

Confirm your selections and click **Configure** to proceed.

When the configuration has completed successfully click **Close**.



As a test you can open a web browser and go to **http://<your server name>/certsrv** to confirm that the certificate services web page loads.



Install a Virtual Machine for Exchange Server 2016

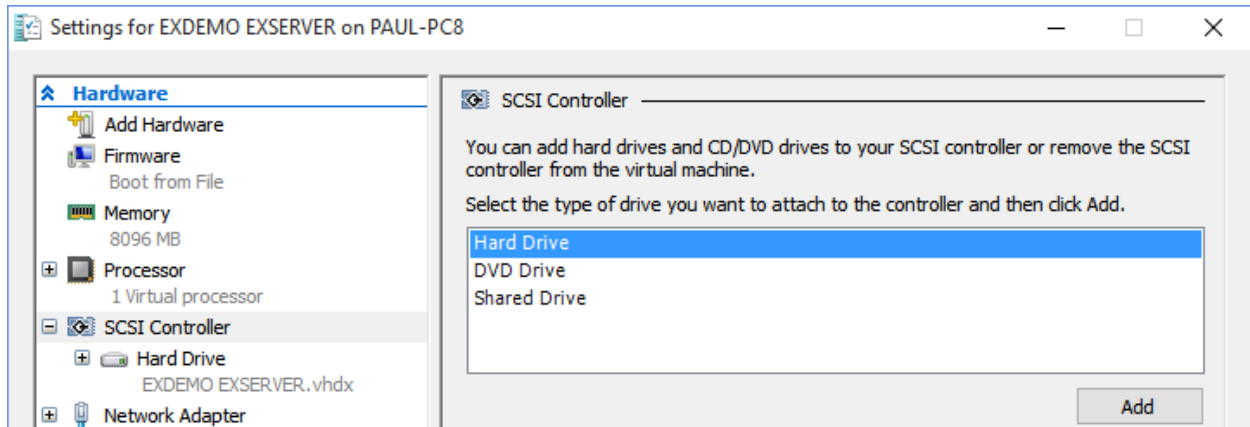
With Active Directory set up and, optionally, a CA installed we can proceed with installation of the first Exchange Server 2016 server.

Using the same steps shown when creating the domain controller we create a new differencing disk and virtual machine, allocating at least 8GB of memory to the virtual machine if possible,

and perform the initial configuration steps for the server name, IP address, and Windows Update configuration. We can also join the server to the Active Directory domain.

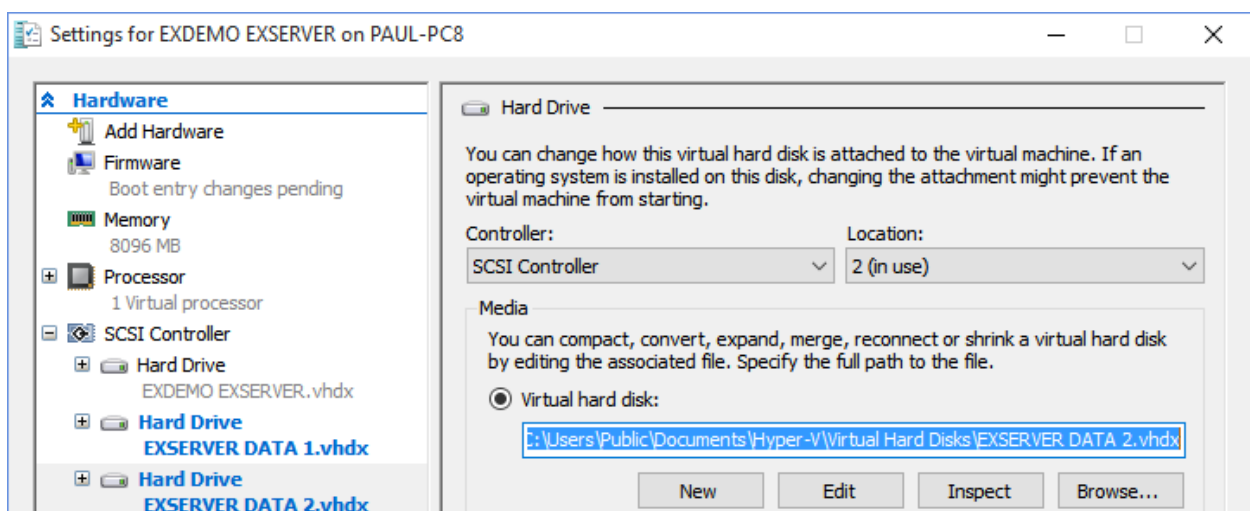
We should also add some more storage to the server to host our Exchange Server 2013 mailbox databases and transaction log files.

In the Settings of the virtual machine select the virtual SCSI Controller and add two more hard drives of equal size.



Create a new hard drive, this time as a **Dynamically Expanding** drive, which allows you to allocate a reasonable size such as 50Gb without consuming all of that space immediately on your host machine.

Repeat the process so you have two new hard drives added to the virtual machine. One will be used for the Exchange databases, and the other for the transaction log files.



After creating the virtual hard disks open Server Manager on the server, navigate to **File and Storage Services → Volumes → Disks**, and bring the new disks online.

Then create a new volume on each disk, using the full amount of available disk space. Assign drive letter D: to the first volume and give it a label of “Databases”. Assign drive letter E: to the second volume and give it a label of “Logs”.

Installing a Virtual Machine as a Client Machine

When it comes to testing Exchange Server 2016 from the client perspective there are a few approaches you can take:

- Allowing regular user accounts to login to one of the servers, such as the domain controller, and running Outlook from there (not ideal, but it’s only a test lab so it isn’t harmful to do it that way).
- Using only Outlook Web App (also not ideal, as OWA and Outlook connect over different protocols, and we’d like to test out both of them)
- Installing a virtual machine to act as the client workstation (ideal, as long as you have the resources on your host machine to run another VM)

You can install a client machine using either any current version of Windows. We would generally recommend Windows 7 or Windows 10 as they are a little easier to interact with inside an RDP window using keyboard and mouse than Windows 8.

For Outlook itself we recommend installing Outlook 2013 or Outlook 2016 on your client machine.

As a final tip, we recommend using a dynamically expanding virtual hard disk to save on storage space on your host machine, and using dynamic memory for the VM if necessary to allow your entire test lab environment to be running at the same time.

Lab Guide Summary

Having a fully functional test lab environment will be an enormous help to you as you work through this guide and other Exchange Server 2016 training in future.

Create the test lab that you're able to fit within your resources and budget. With so many different options available now for hosting a lab, such as cheap server hardware, powerful laptops and desktops, or even cloud services like Microsoft Azure, a test lab environment is within reach for most people.