
How to Setup and Configure a Windows Server 2016/2012 as a Remote Desktop Session Host Server (Terminal Server).

Notes:

1. The mentioned steps below, can be applied either on a Domain Controller or in a standalone server 2016/2012.

2. If the Terminal services are installed on a server that will act as a Domain Controller also, then first install the Active Directory Domain Service (AD DS) role service and promote the Server to a Domain Controller, before installing the Remote Desktop Session Host (RDSH) role service (Terminal Service).

3. Keep in mind that the below configuration does provide access to RemoteApp programs or the RDWeb site, because the Remote Desktop Connection Broker role service will not be installed.

Step 1. Install Remote Desktop Services on Server 2016/2012.

Step 2. Activate the Remote Desktop License Server.

Step 3. Install Licenses on the Remote Desktop License Server.

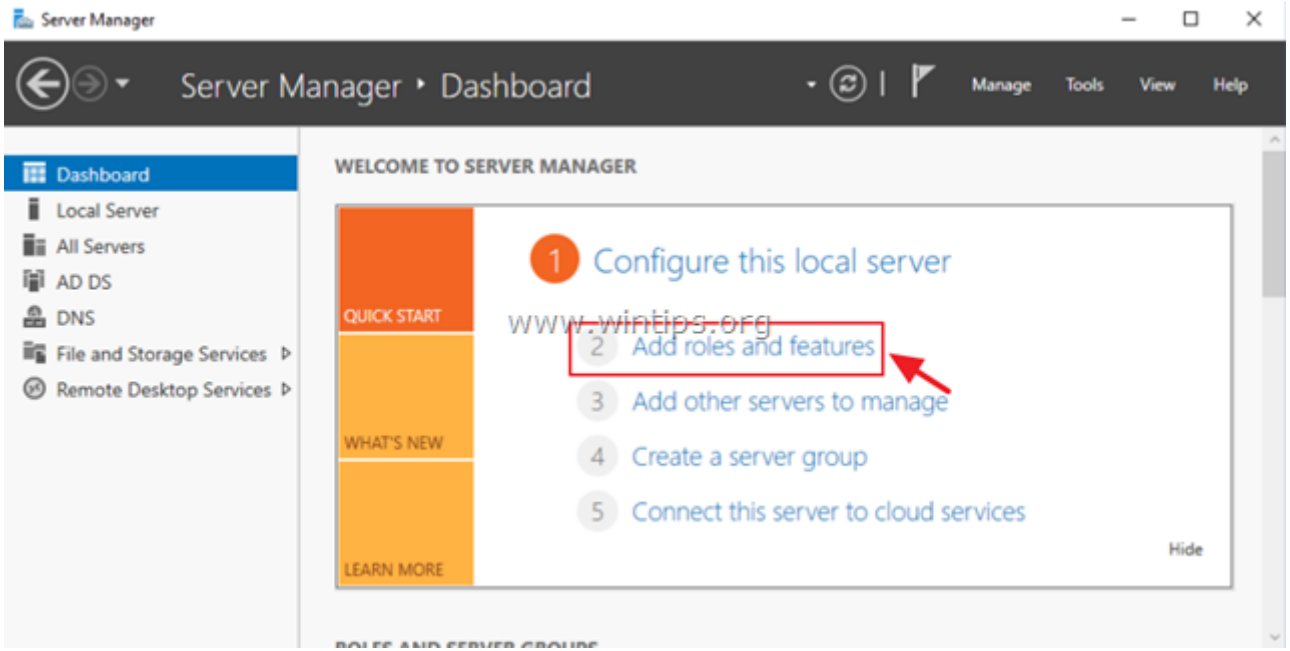
Step 4. Configure RD Session Host role to use the local Remote Desktop Licensing server & Set the Remote Desktop licensing mode.

Step 5. Add RD Clients (Users) to the Remote Desktop Users Group.

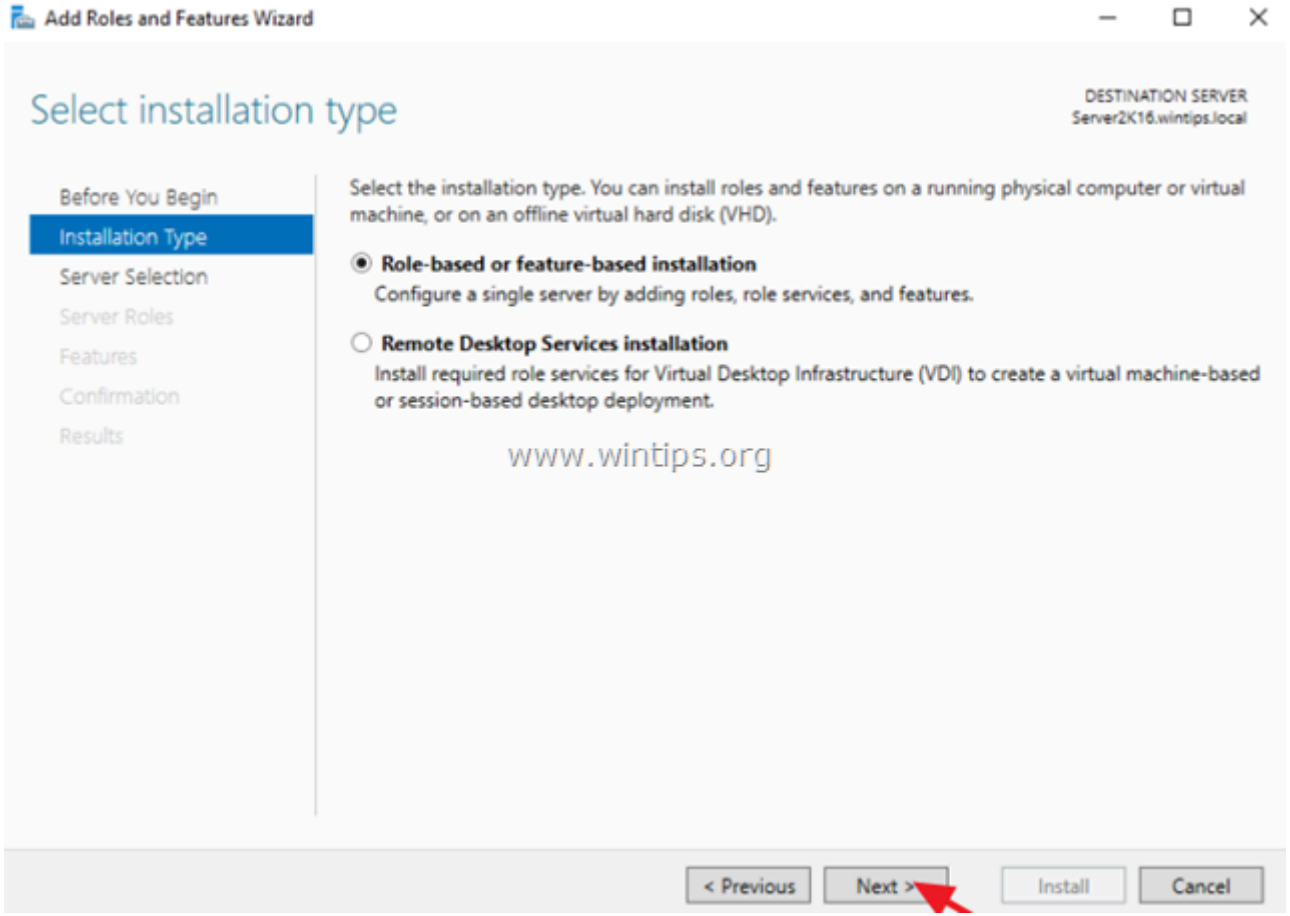
Step 6. Allow the log on through remote desktop Services.

Step 1. Install Remote Desktop Licensing and Remote Desktop Session Host role services.

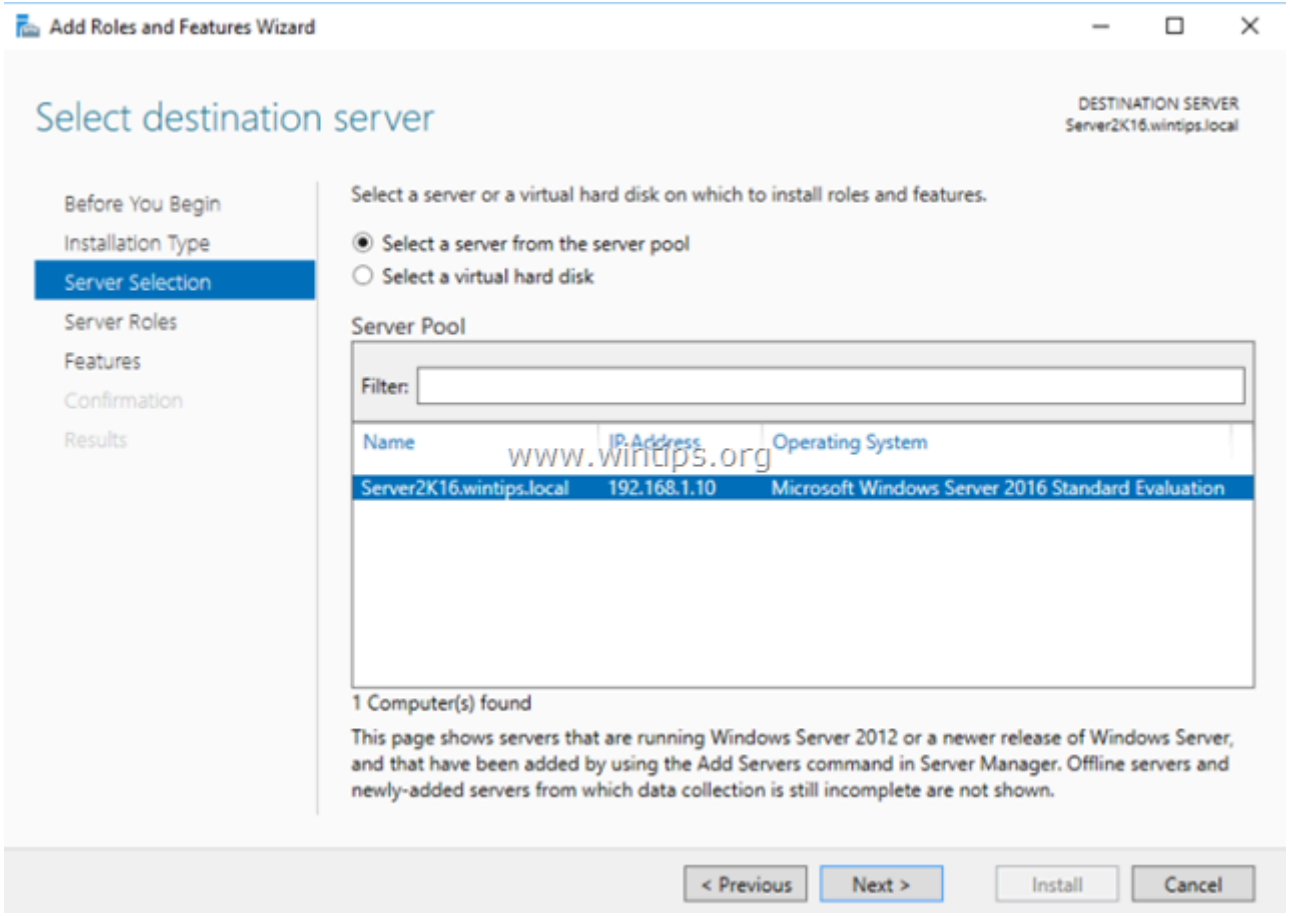
1. Open 'Server Manager' and click on **Add Roles and Features**.



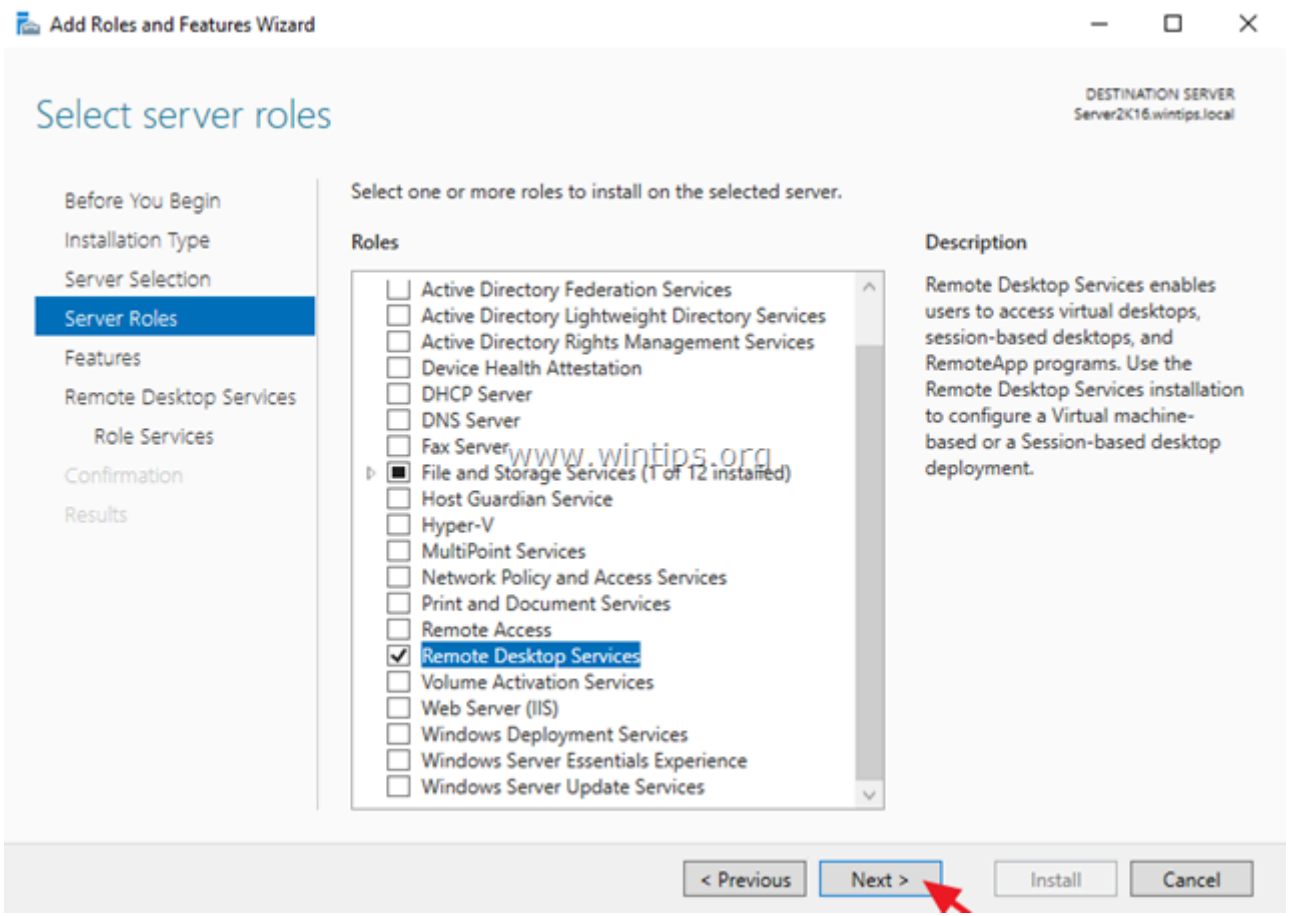
2. At the first screen of 'Add Roles and Features wizard' leave the **Role-based or feature-based installation** option and click **Next.**



3. At the next screen, leave the default option "**Select server from the server pool**" and click **Next.**

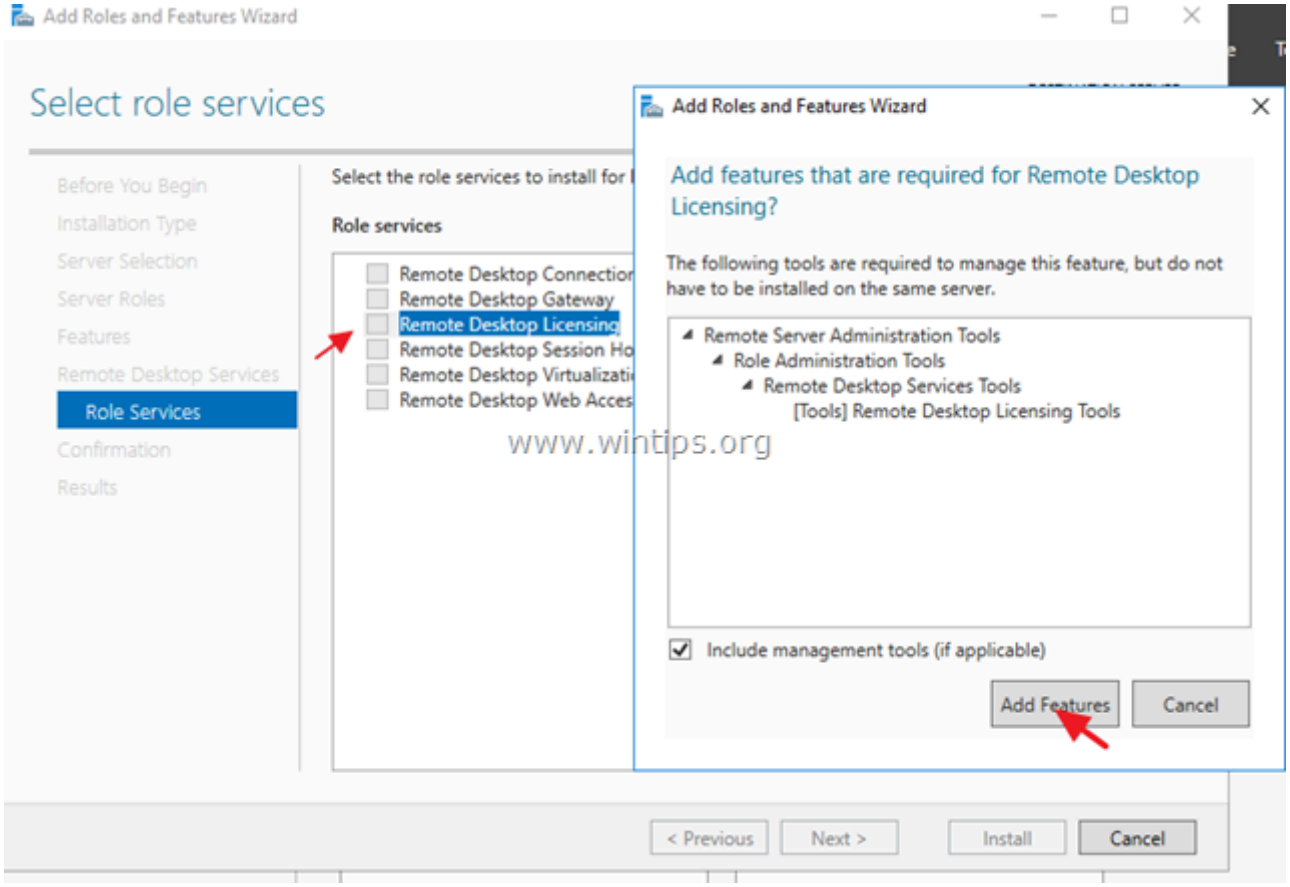


4. Select the **Remote Desktop Services** and click **Next**.

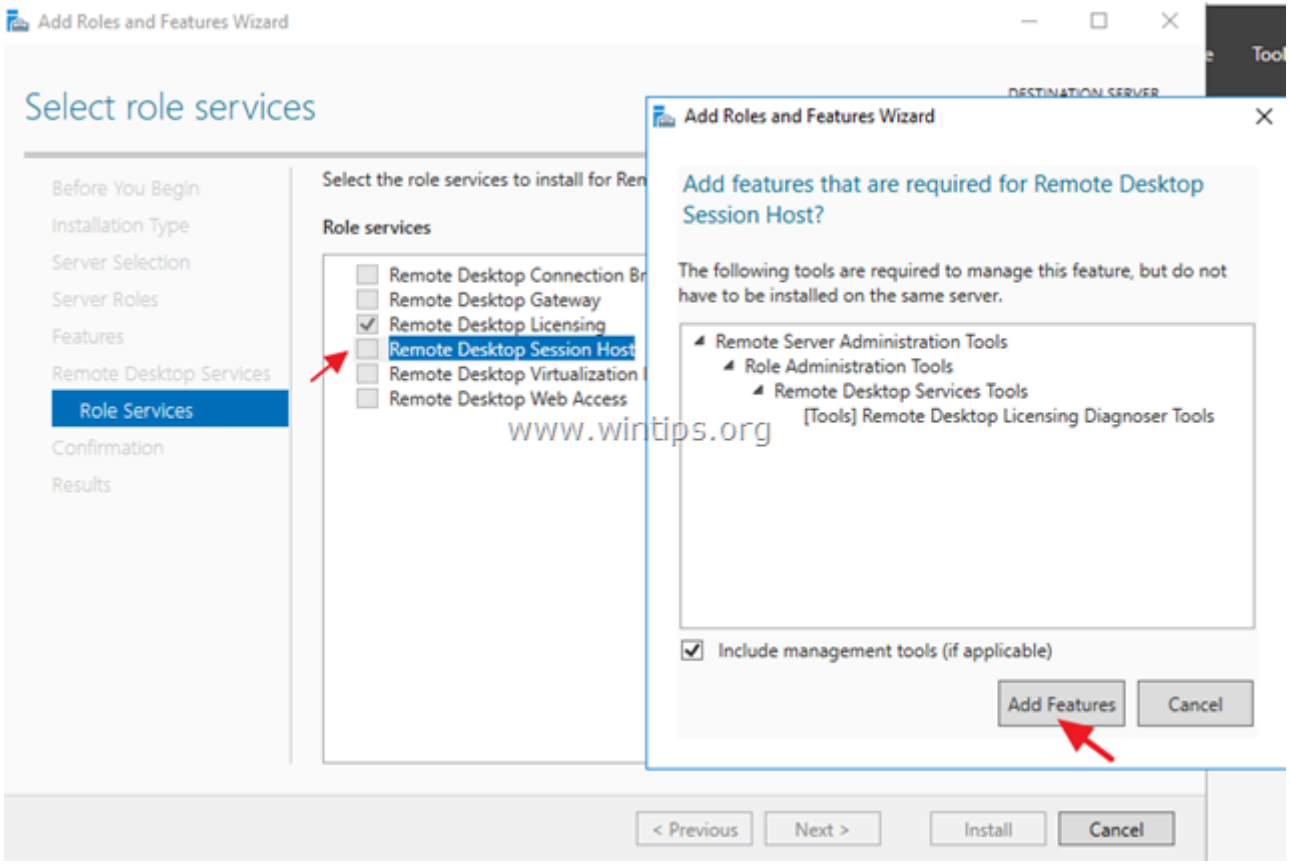


5. Leave the default settings and click **Next** at **Features** and **Remote Desktop Services** screens.

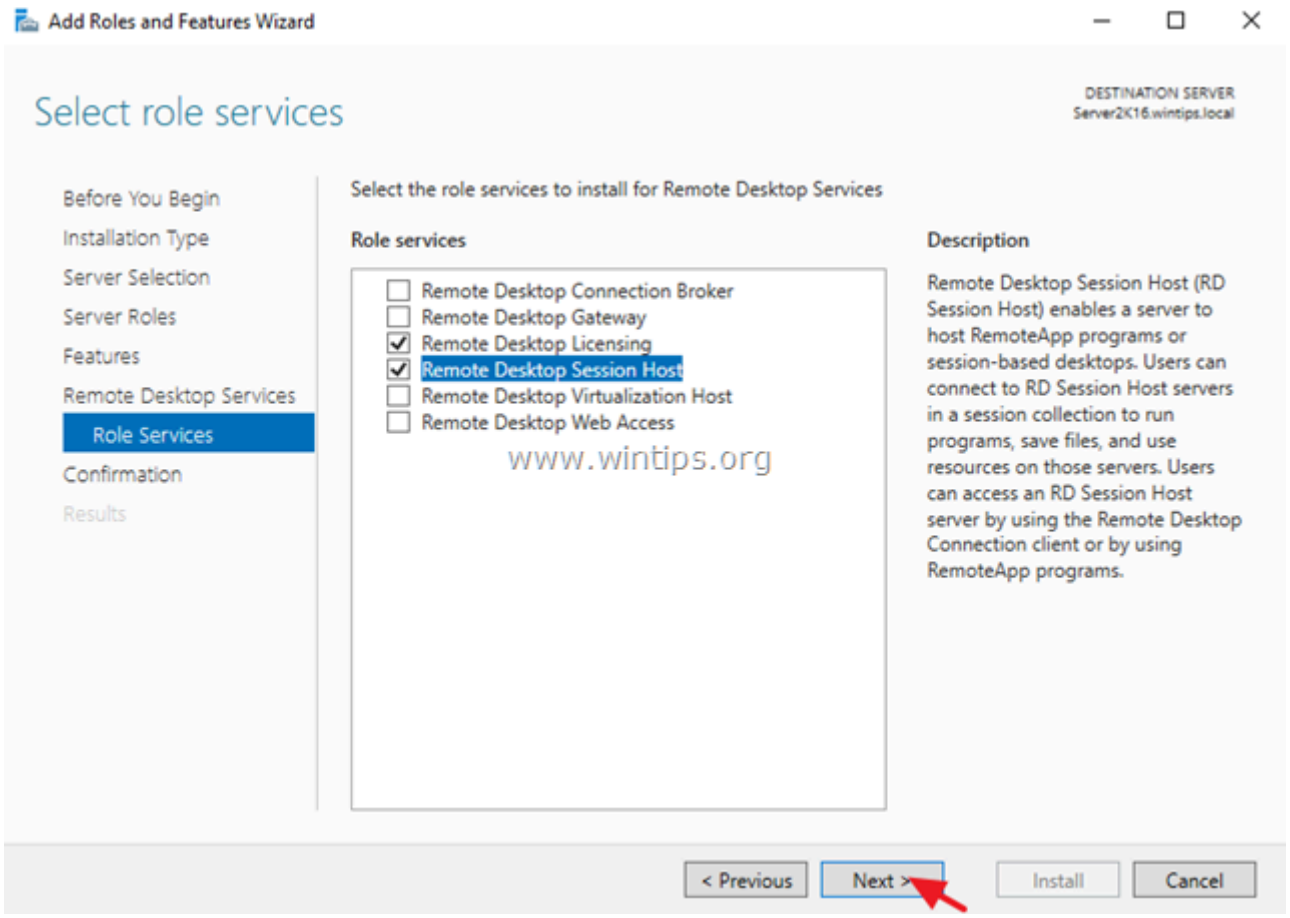
6. At **Role Services** screen, select the **Remote Desktop Licensing** role service and then click **Add Features**.



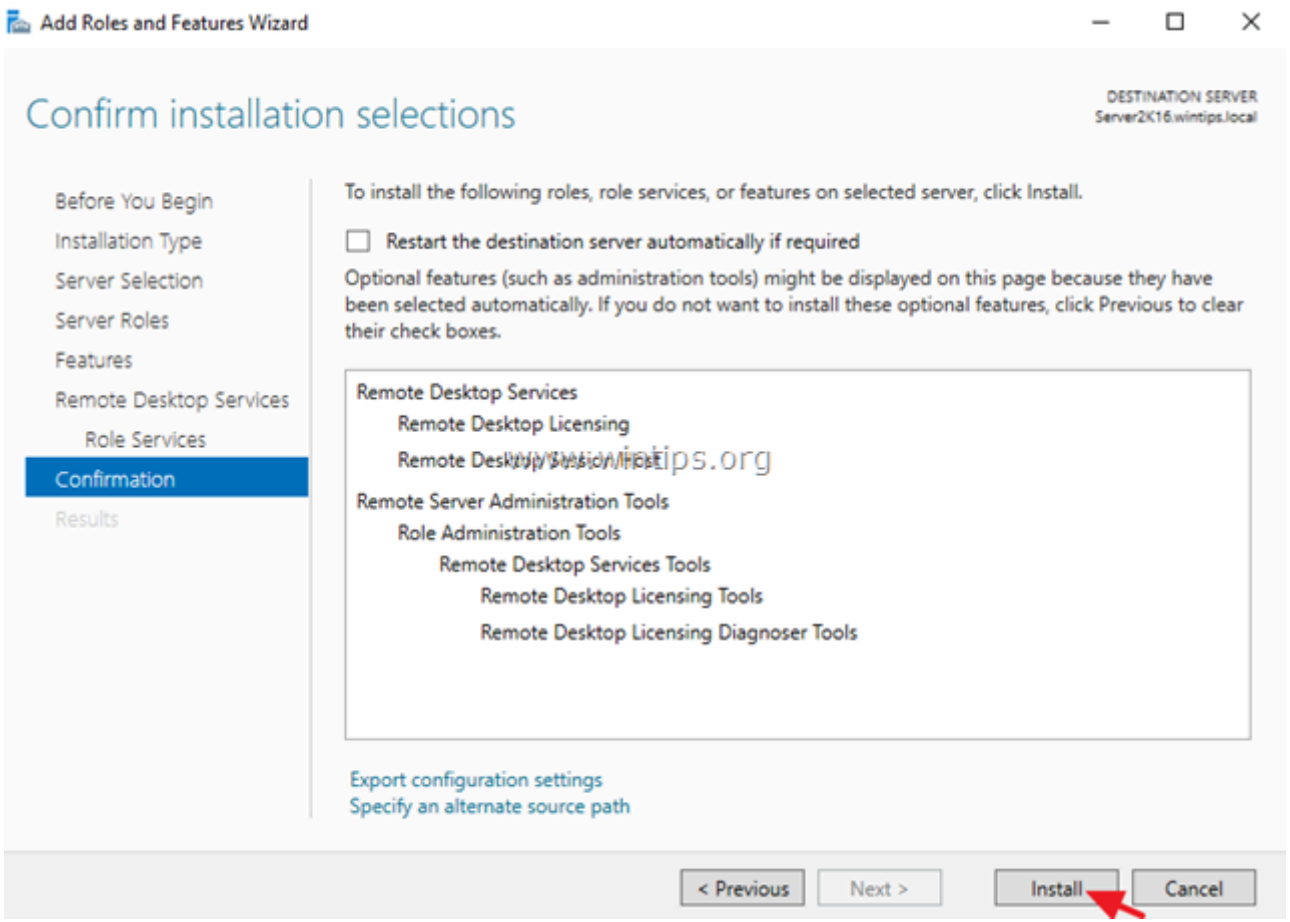
7. Then select the **Remote Desktop Session Host** role service and click Add Features again.



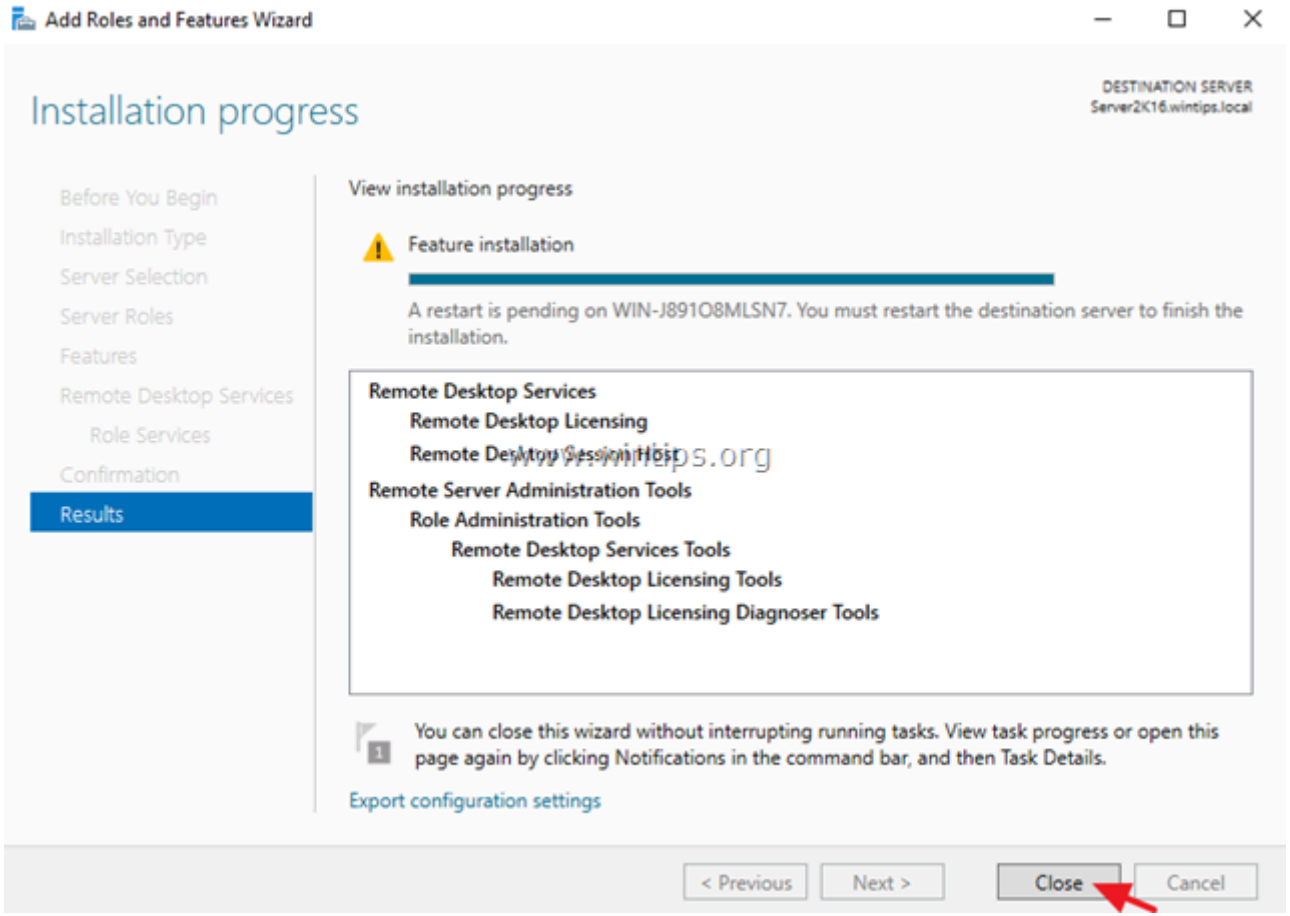
8. When done, click **Next** to continue.



9. Finally click Install to install the Remote Desktop Services: Remote Desktop Licensing and Remote Desktop Session Host.



10. When the installation is completed **close** the 'Add Roles and Features Wizard' and **restart** your server.

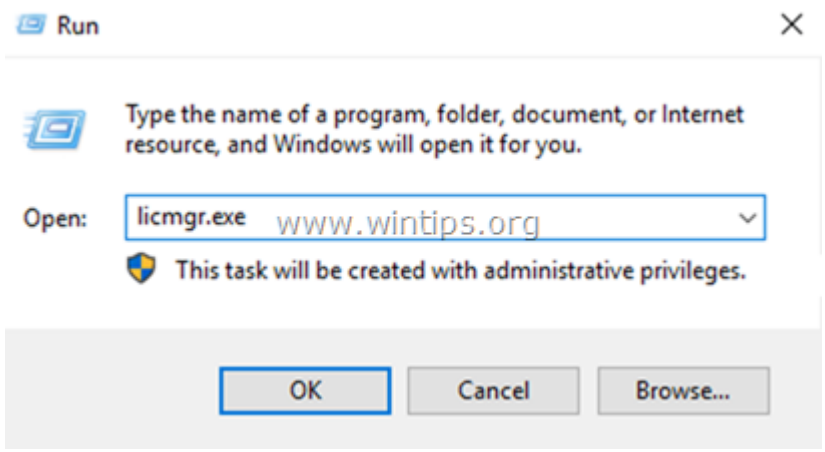


Step 2. Activate the Remote Desktop License Server.

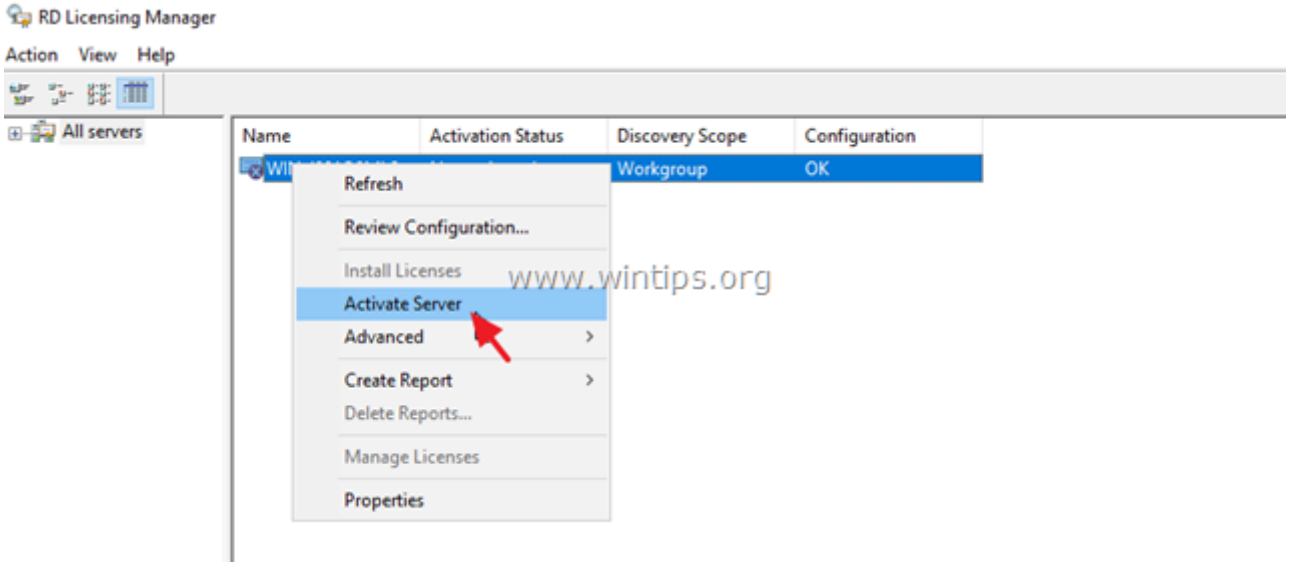
1. Simultaneously press the **Windows** + **R** keys to open run command box.

2. Type **licmgr.exe** and press **Enter** to open the RD Licensing Manager *

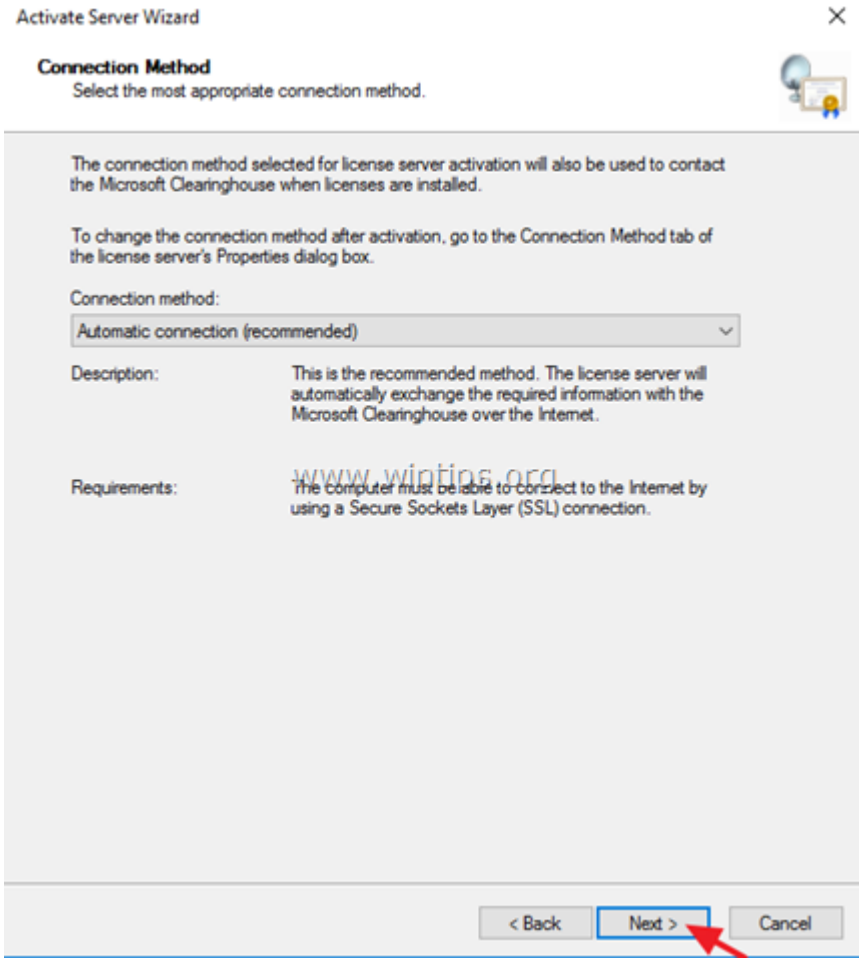
* Note: Alternately, you can launch the RD Licensing Manager, from Control Panel –> Administrative Tools –> Remote Desktop Services –> Remote Desktop Licensing Manager.



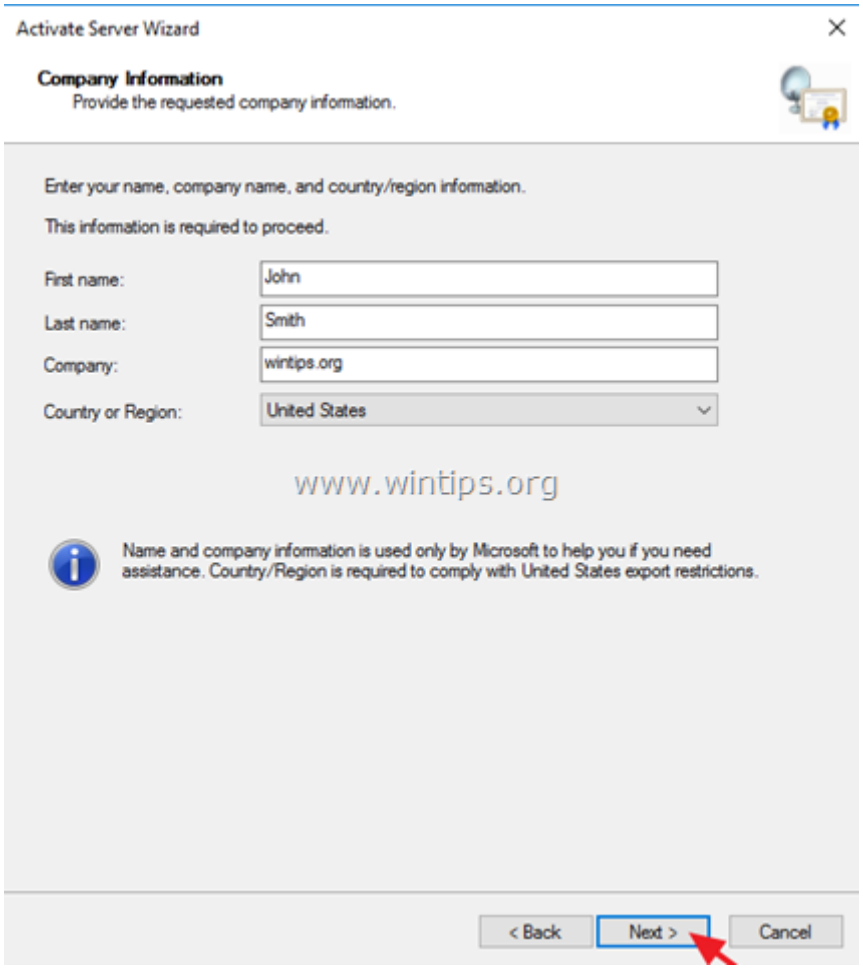
3. At the right pane, right click on the server name and select **Activate Server.**



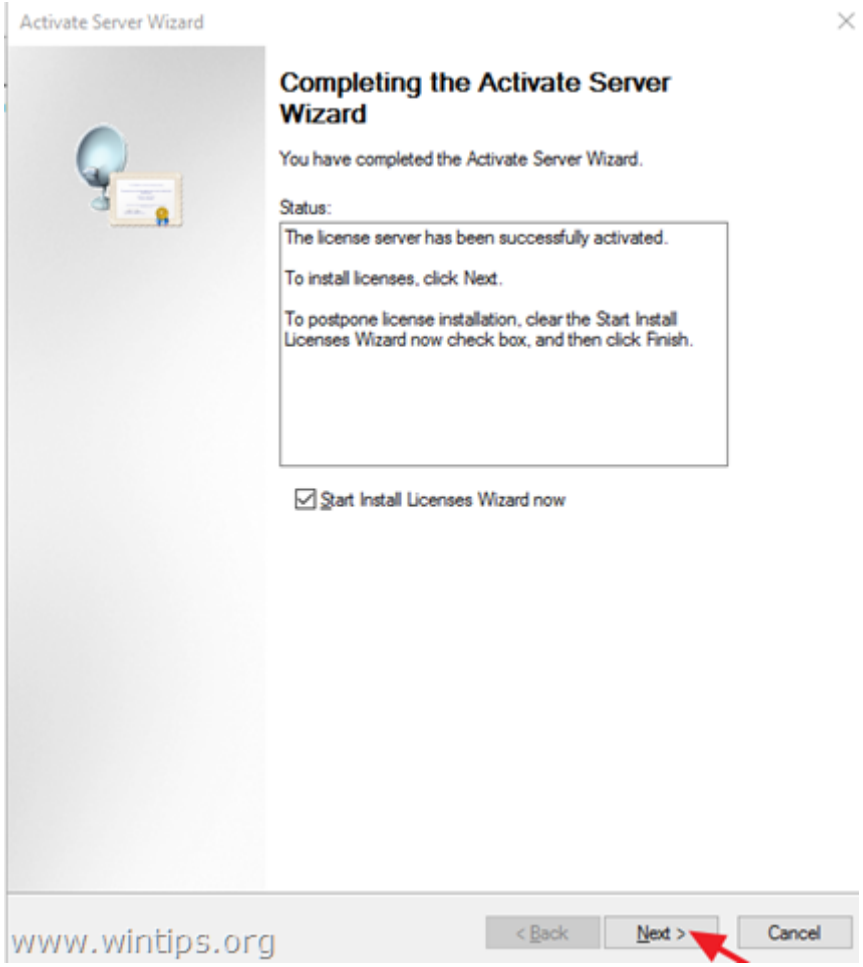
4. Click **Next** at the Welcome screen and then click **Next** again at Connection method options.



5. At 'Company Information' window, fill the required fields and click **Next** twice to activate your License Server.



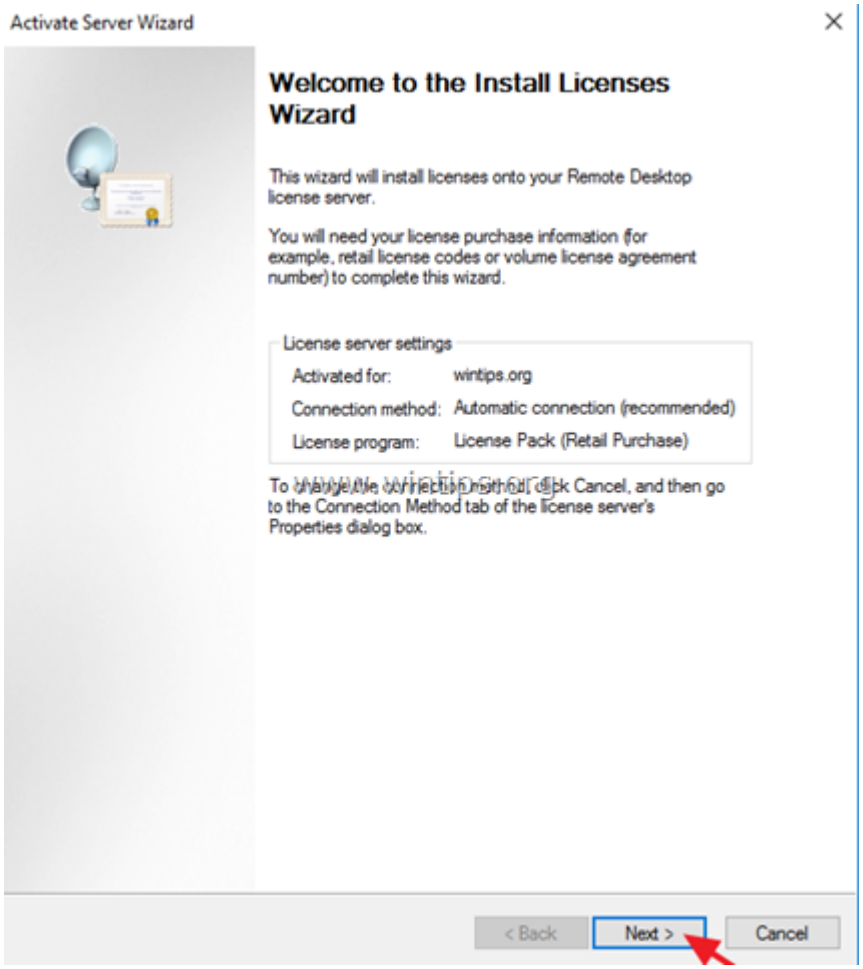
6. When the activation is completed, leave checked the 'Start Install Licenses Wizard' checkbox and click **Next**.



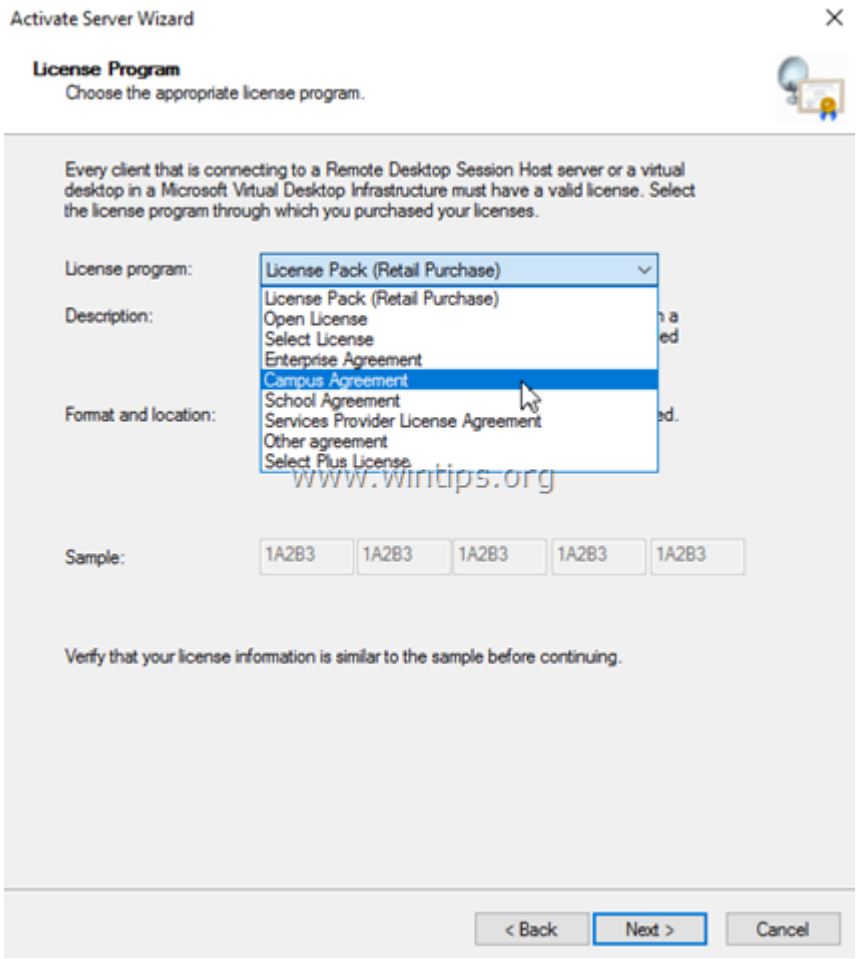
7. Continue to next step.

Step 3. Install Licenses on the Remote Desktop License Server.

1. At 'Welcome to the install licenses wizard', click **Next**



2. On the **License Program** page, select the appropriate program through which you purchased your RDS CALs, and then click **Next**.



3. According the **License Program** you selected on the previous page, type either the license code or the agreement number provided when you purchased your RDS CALs and then click **Next**.

4. On the **Product Version and License Type** page, select the appropriate product version, license type, and the quantity of the RDS CALs based on your RDS CAL purchase agreement, and then click **Next**.

5. When the RDS CALs installed on the server, click **Finish**. *

TIP: If you cannot activate the RDS Server automatically, then try to activate it using the Web Browser or via Telephone. To do that:

a. Right-click on the Server's name and select ***Properties.***

b. ***Change the Connection Method*** *to* ***Web Browser*** *or to* ***Telephone****. When done, click* ***OK.***

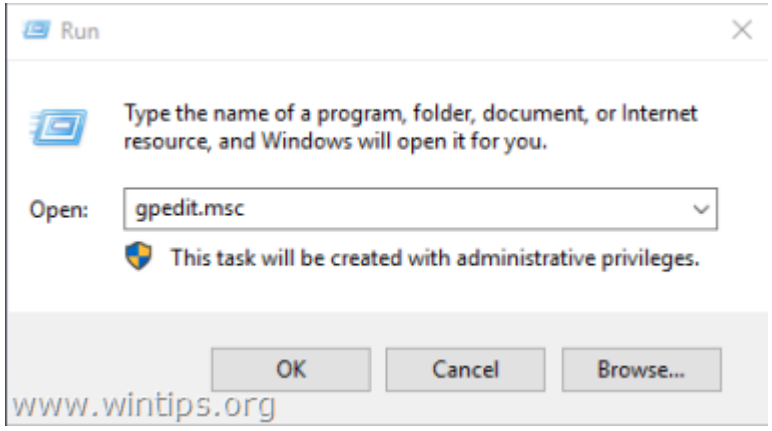
c. Finally, right click on the server name, select ***Activate Server*** *and follow the onscreen instructions to completed the activation.*

Step 4. Configure RD Session Host role to use the local Remote Desktop Licensing server & Set the Remote Desktop licensing mode.

1. Open Group Policy Editor. To do that:

1. Simultaneously press the ***Windows*** + ***R*** *keys to open run command box.*

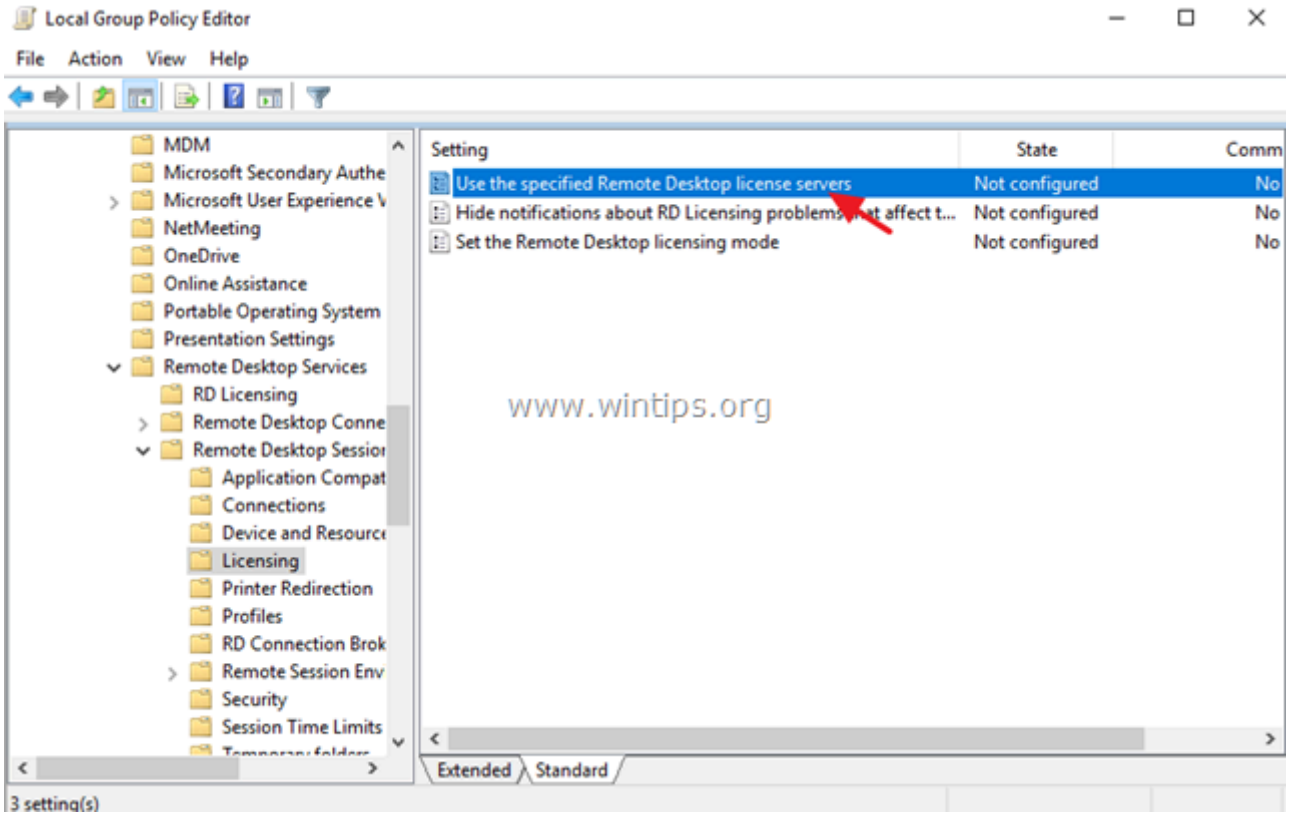
2. Type ***gpedit.msc*** *and press* ***Enter****.*



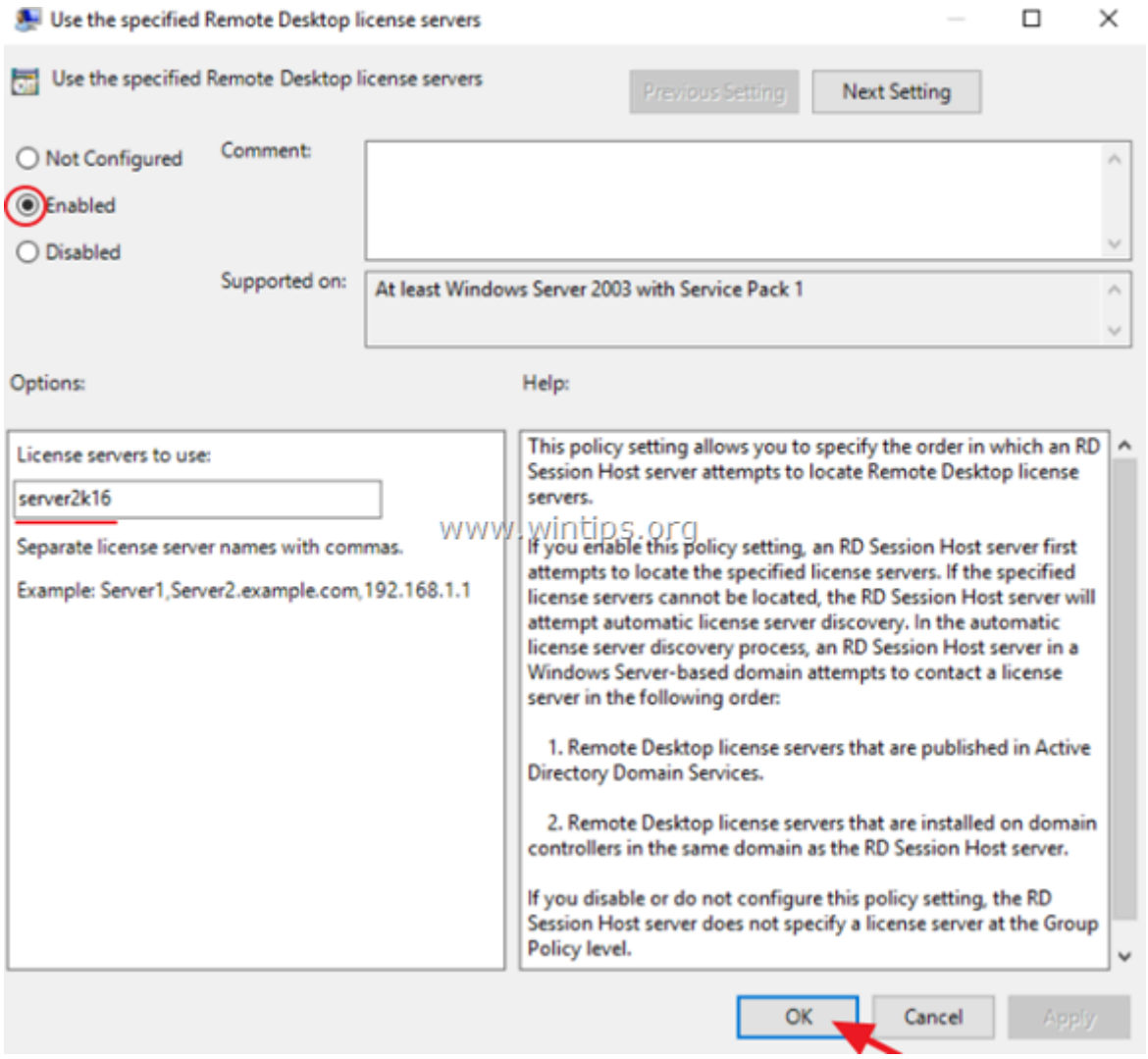
2. In Group Policy Editor navigate to:

- **Computer Configuration\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\ Licensing**

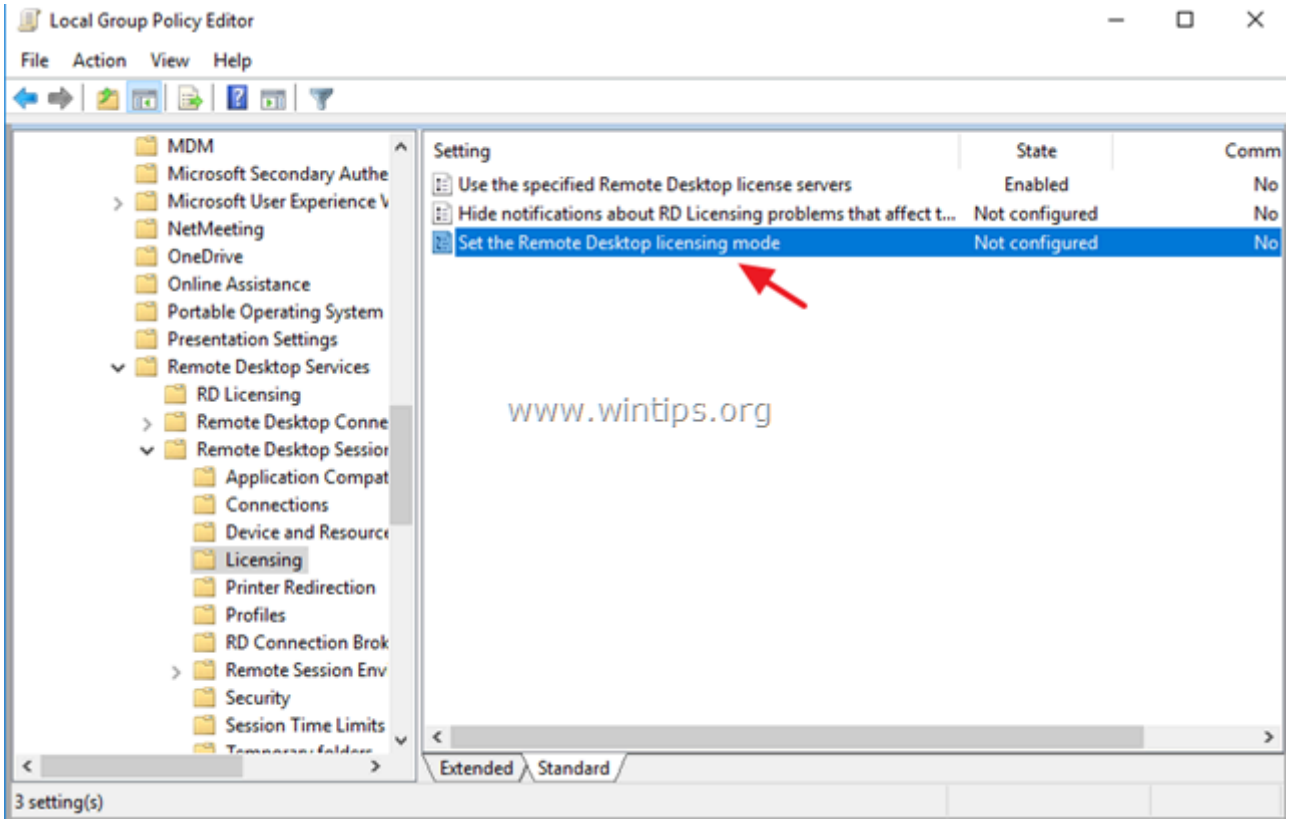
3. At the right pane, double click at **Use the specified Remote License Servers**.



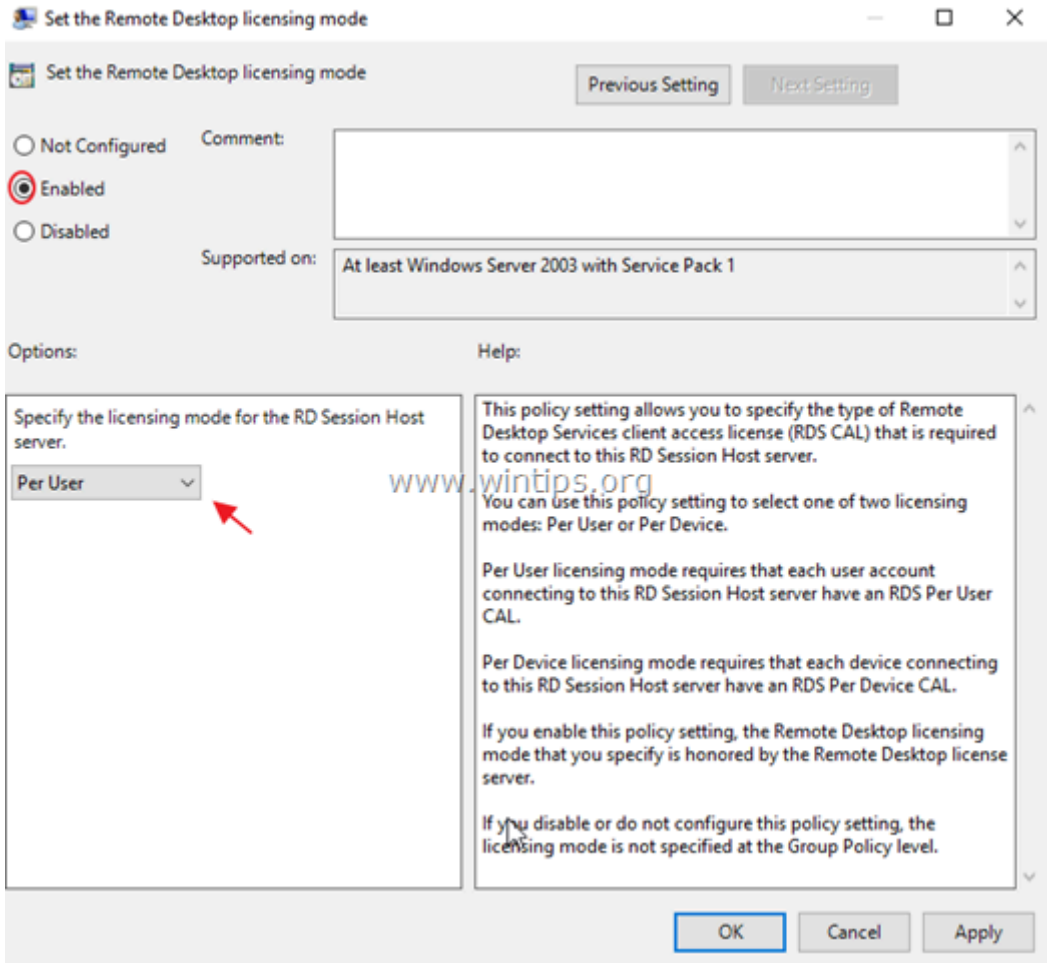
4. Click **Enabled**, and then at '**License server to use**' field, type the RDS license server name and click **OK**.



5. Then open the **Set the Remote Desktop licensing mode** setting.

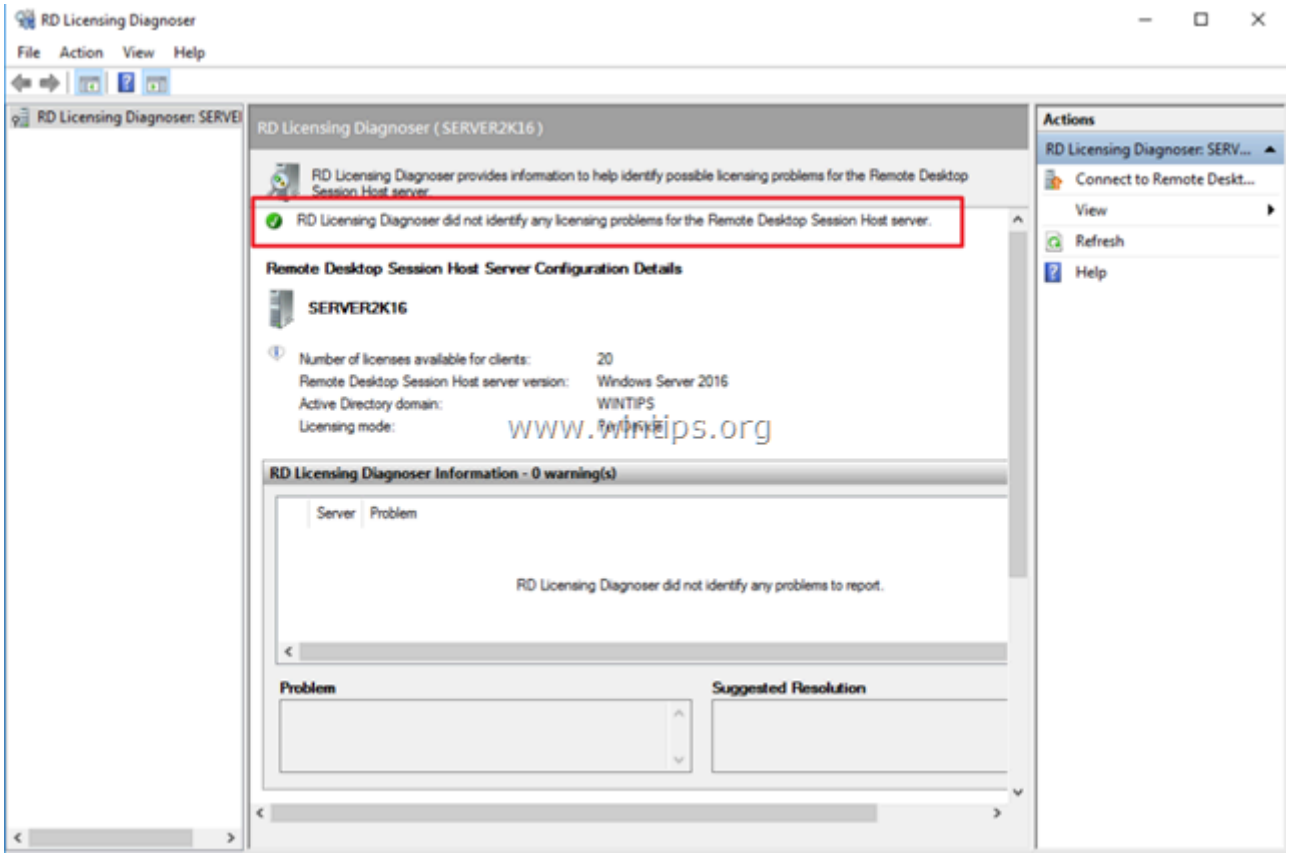


6. Click **Enabled** and then specify the licensing mode (Per User or Per Device) for the RDS host server and then click **OK** again.



7. Close Group Policy Editor.

8. Verify the RD Licensing configuration, by going to: Windows Control Panel –> Administrative Tools –> Remote Desktop Services –> RD Licensing Diagnoser.

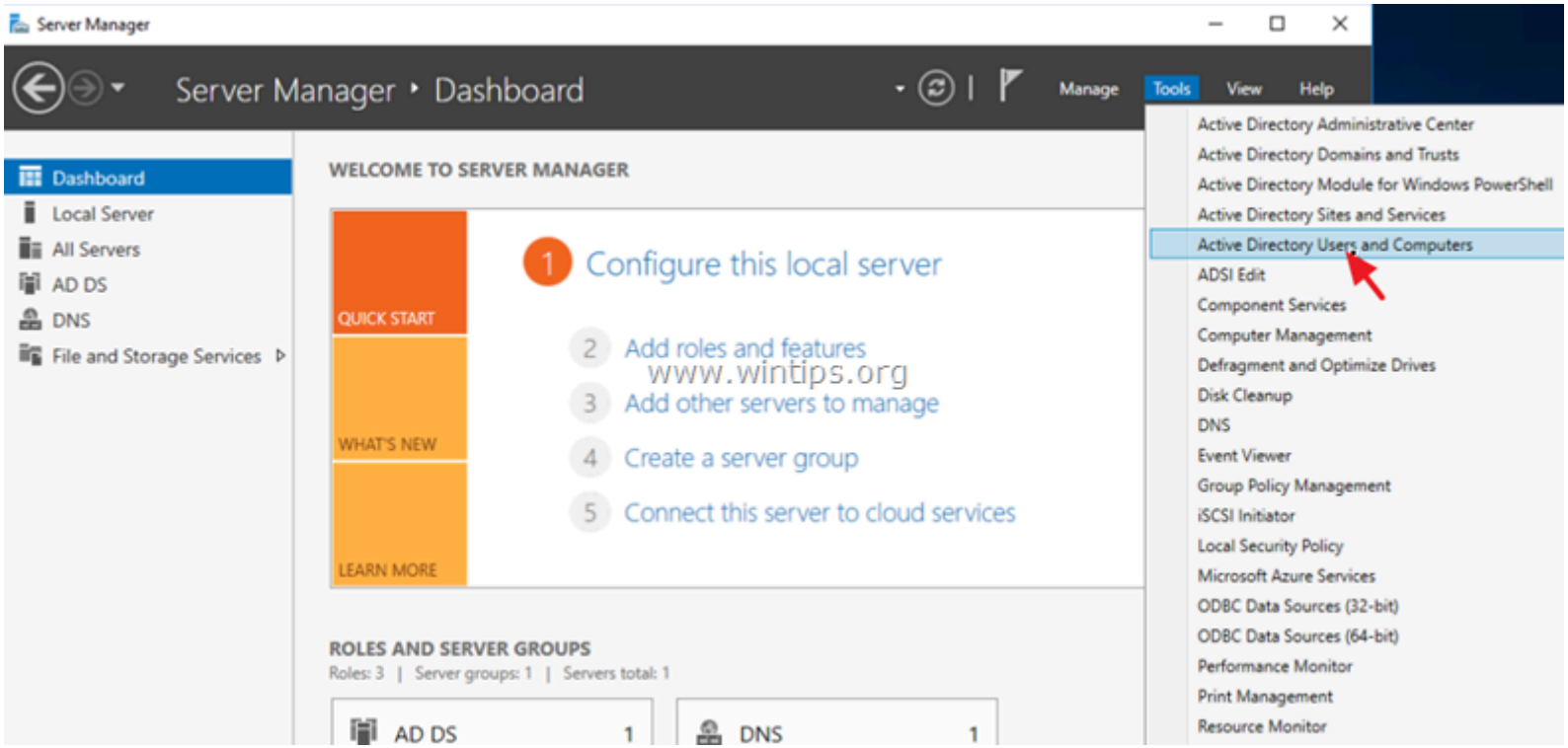


Step 5. Add RD Clients (Users) to the Remote Desktop Users Group.

1. Open **Server Manager**.

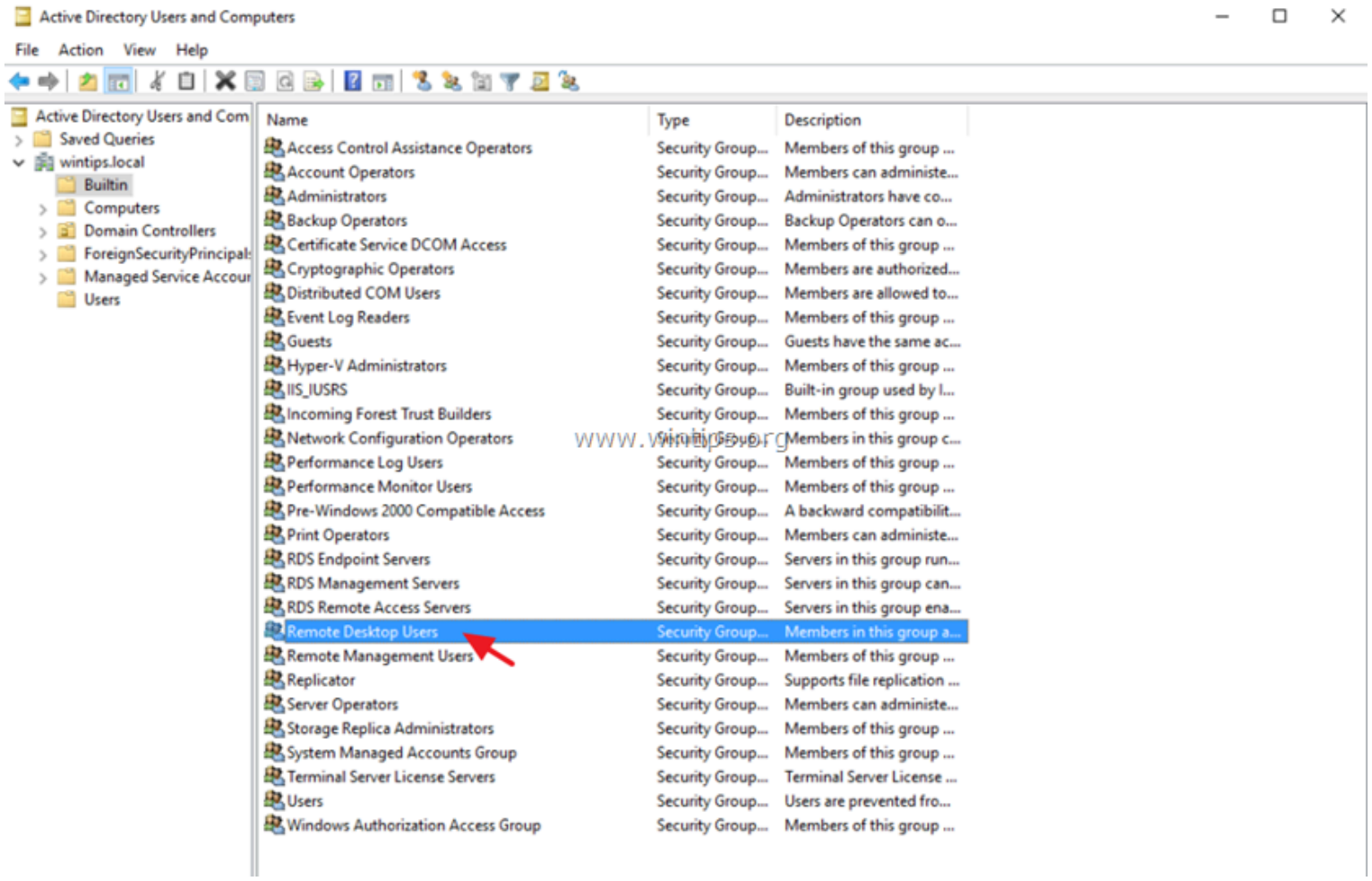
2. From **Tools** menu, select **Active Directory Users and Computers**. *

* Note: If the RD Session Host Service is not installed on the Domain Controller, use the 'Local Users and Groups' snap-in or the 'Remote' tab in the 'System Properties' on the RDS host server, to add the remote desktop users.

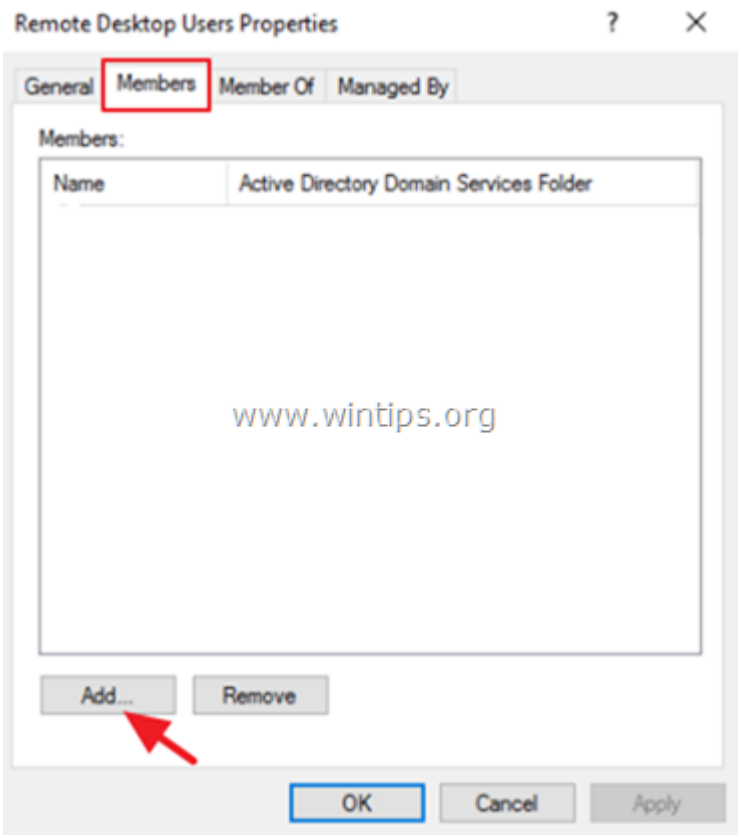


3. Double click at your domain on the left and then select **Builtin.**

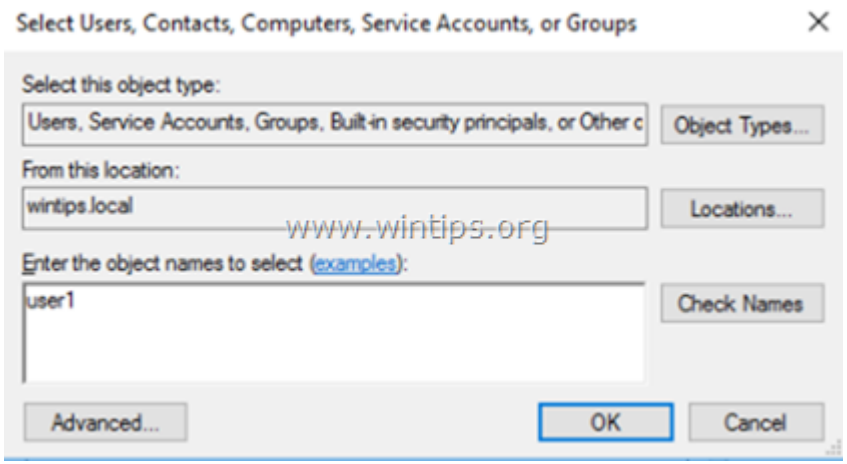
4. Open **Remote Desktop Users** on the right pane.



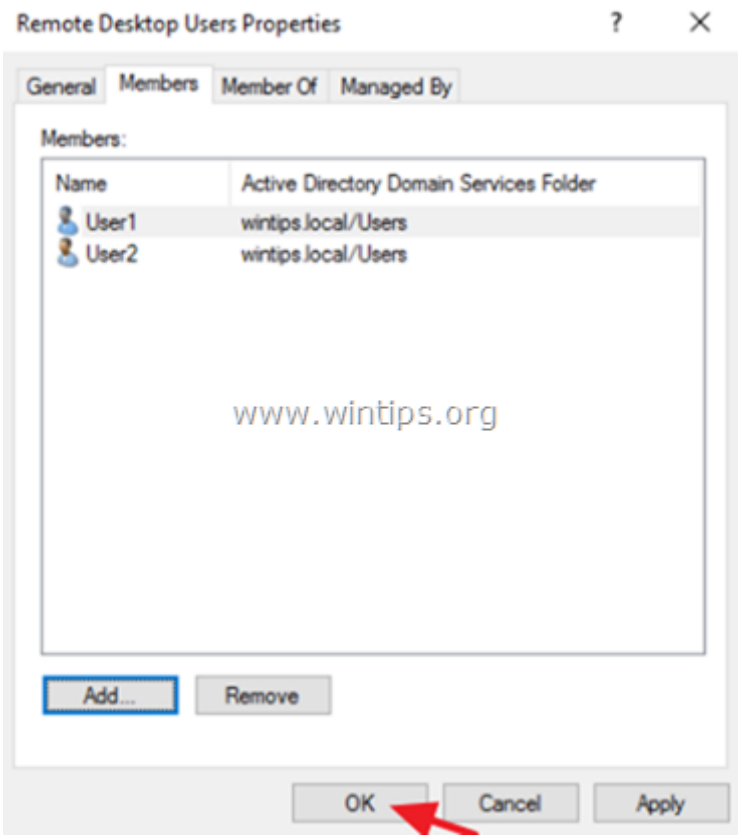
5. At **Members** tab, click **Add**.



6. Type the name(s) of the users that you want to give Remote access to the RDS Server and click **OK**.



7. After selecting the remote desktop users, click **OK** again to close the window.



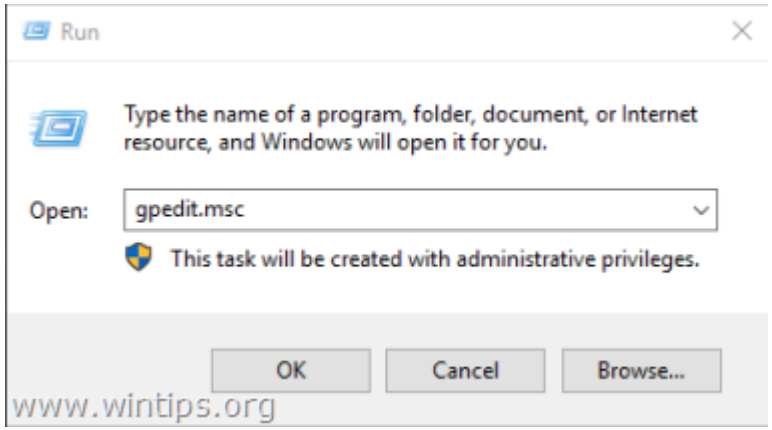
8. Continue to **step-6** below.

Step 6. Allow the log on through remote desktop Services.

1. Open the Local Group Policy Editor. To do that:

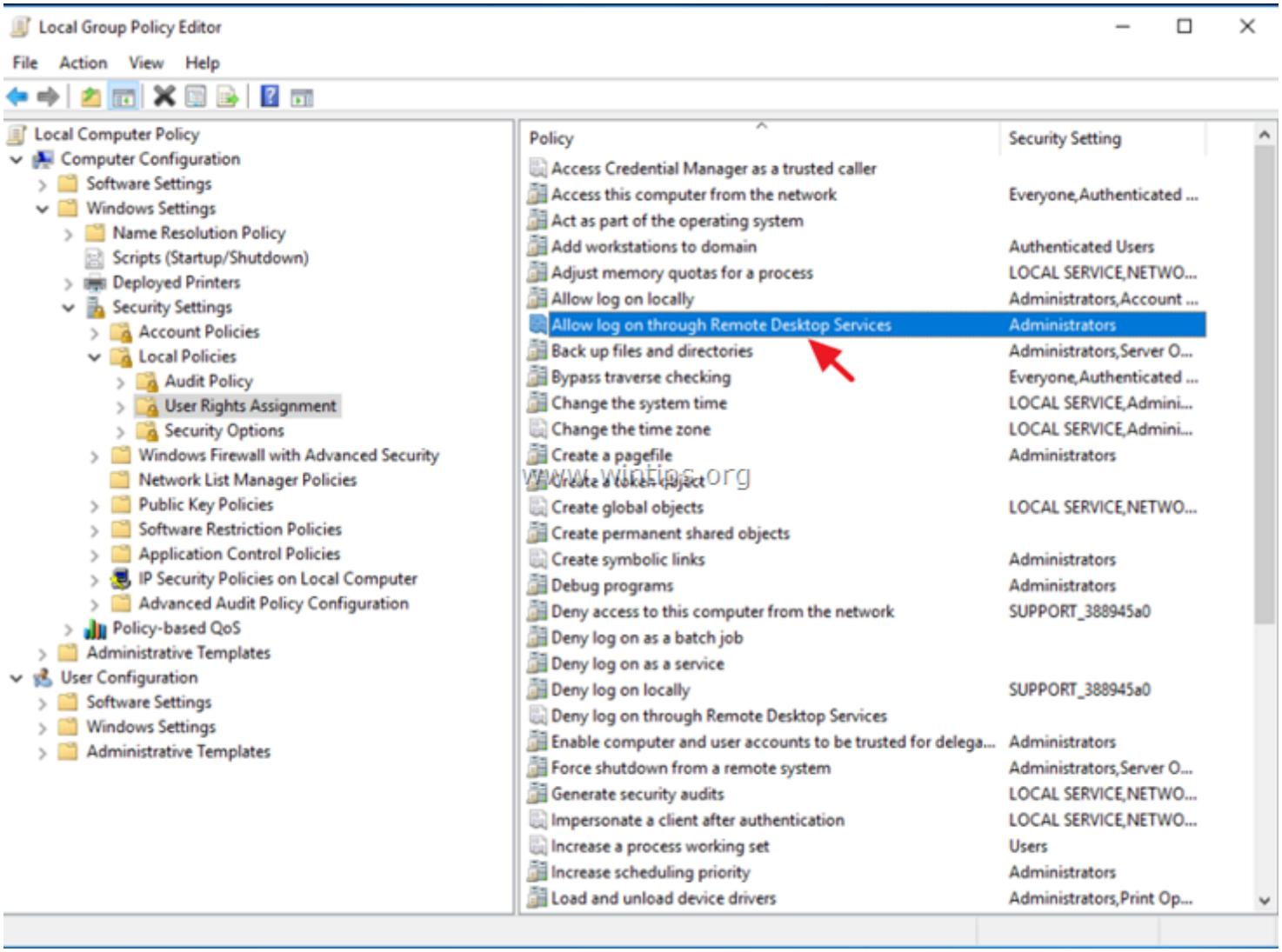
1. Simultaneously press the ***Windows*** + ***R*** *keys to open run command box.*

2. Type ***gpedit.msc*** *and press* ***Enter****.*

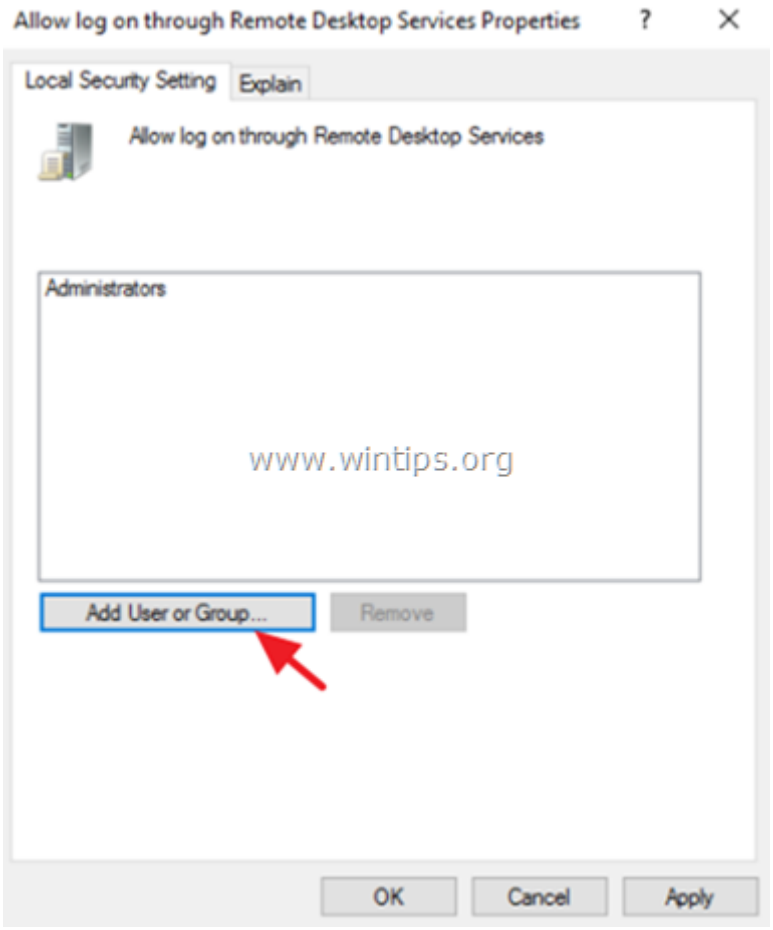


2. In Group Policy Editor navigate to: **Computer Configuration** > **Windows Settings** > **Security Settings** > **Local Policies** > **User Rights Assignment**.

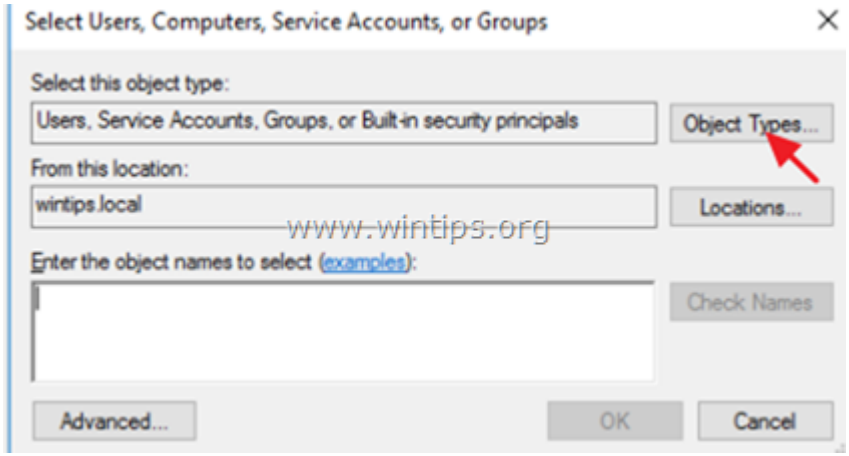
3. At the right Pane: double click at **Allow log on through Remote Desktop Services.**



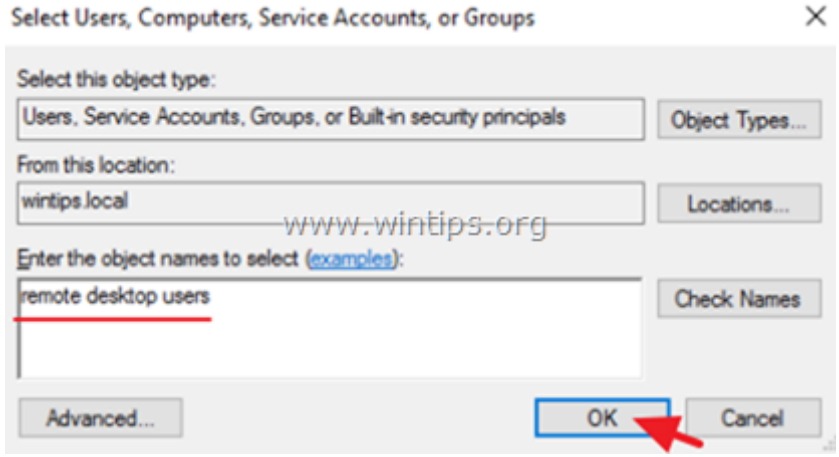
4. Click **Add User or Group**.



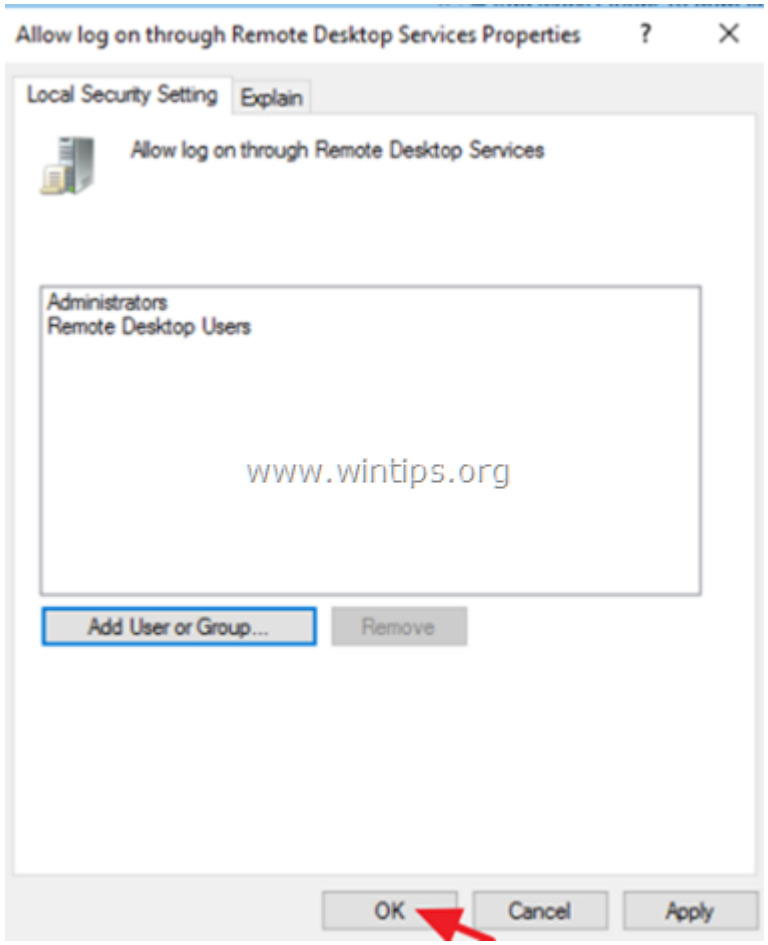
5. Click **Object Types**, check all the available objects (Users, Groups, & Built-in security principals) and then click **OK.**



6. Type **remote desktop users** and then click **OK**.



7. Finally click **OK** again and **close** Group Policy Editor.

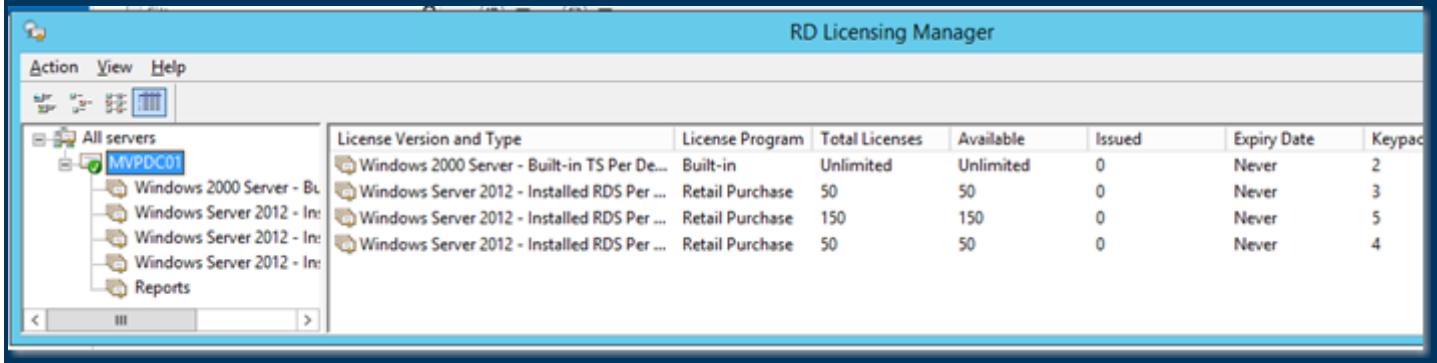


8. Now you 're ready to connect to the Remote Desktop Session Host Server 2016/2012 from any Remote desktop client.

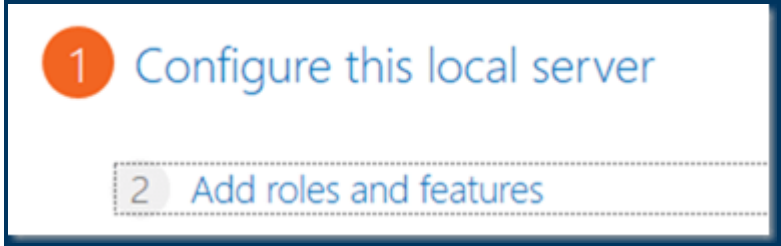
----------X----------

Step by Step Server 2016 Remote Desktop Services QuickStart Deployment [Application]

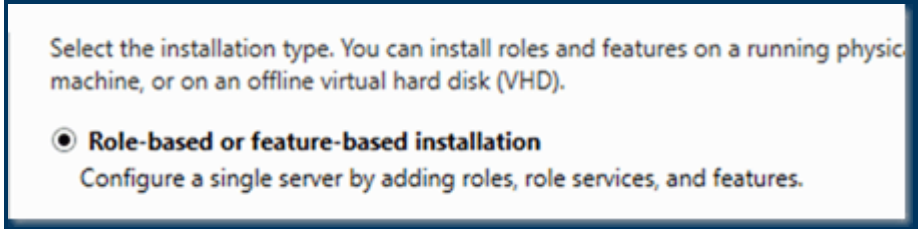
My DC is running the License services and this is also my broker server.



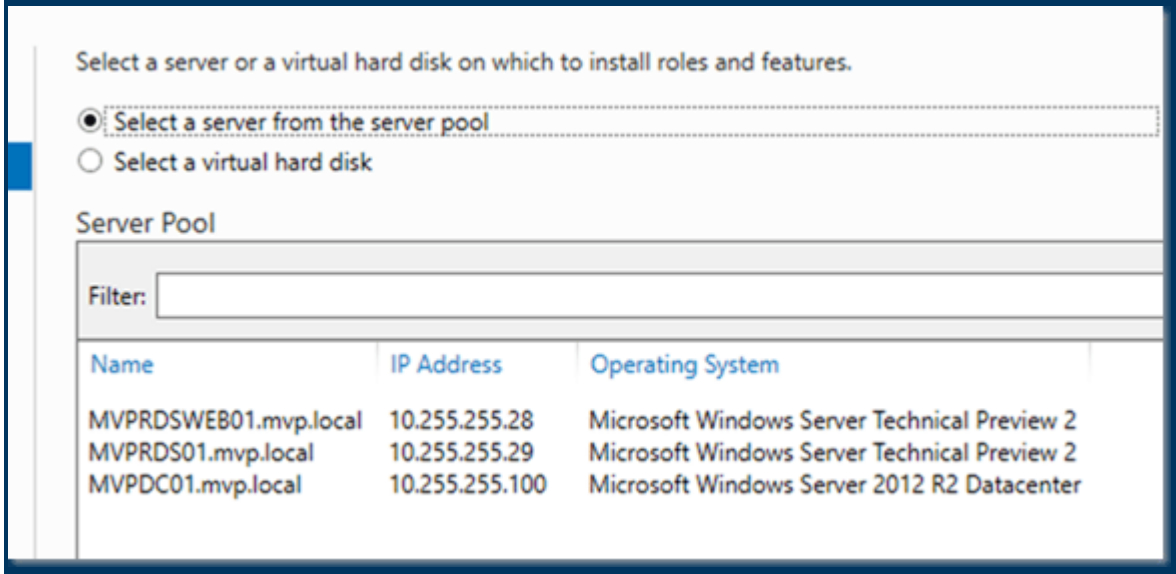
Doing this setup is in two parts One add Roles and Second the RDS setup.

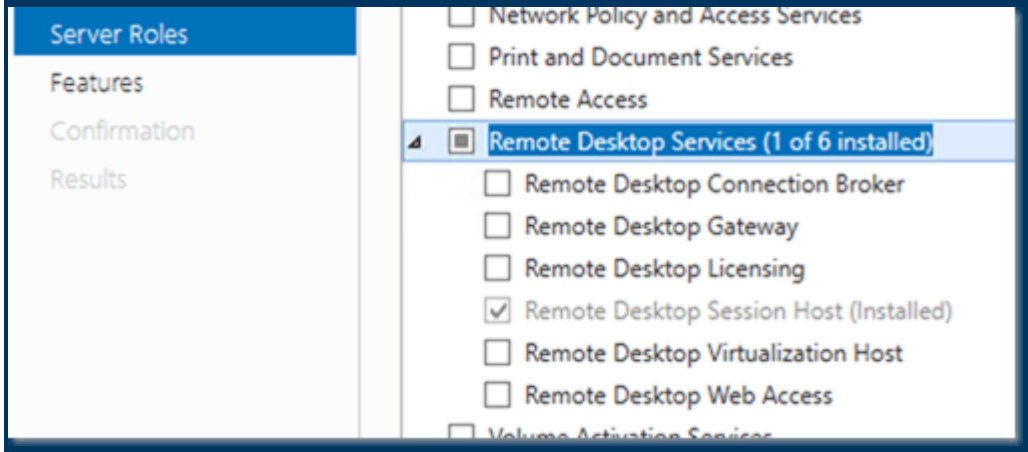


Adding the Roles to my DC and adding all the servers in the all server filter in the server manager of the DC.

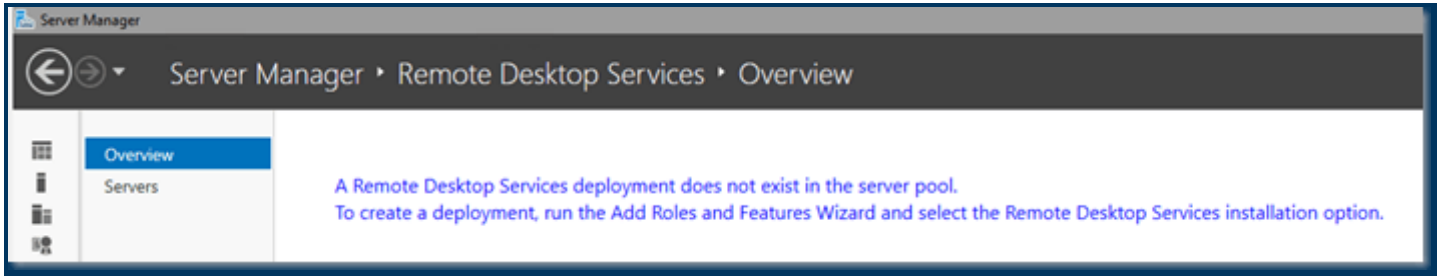


Selecting the Server that holds the Remote Desktop Session host (mvprds01)



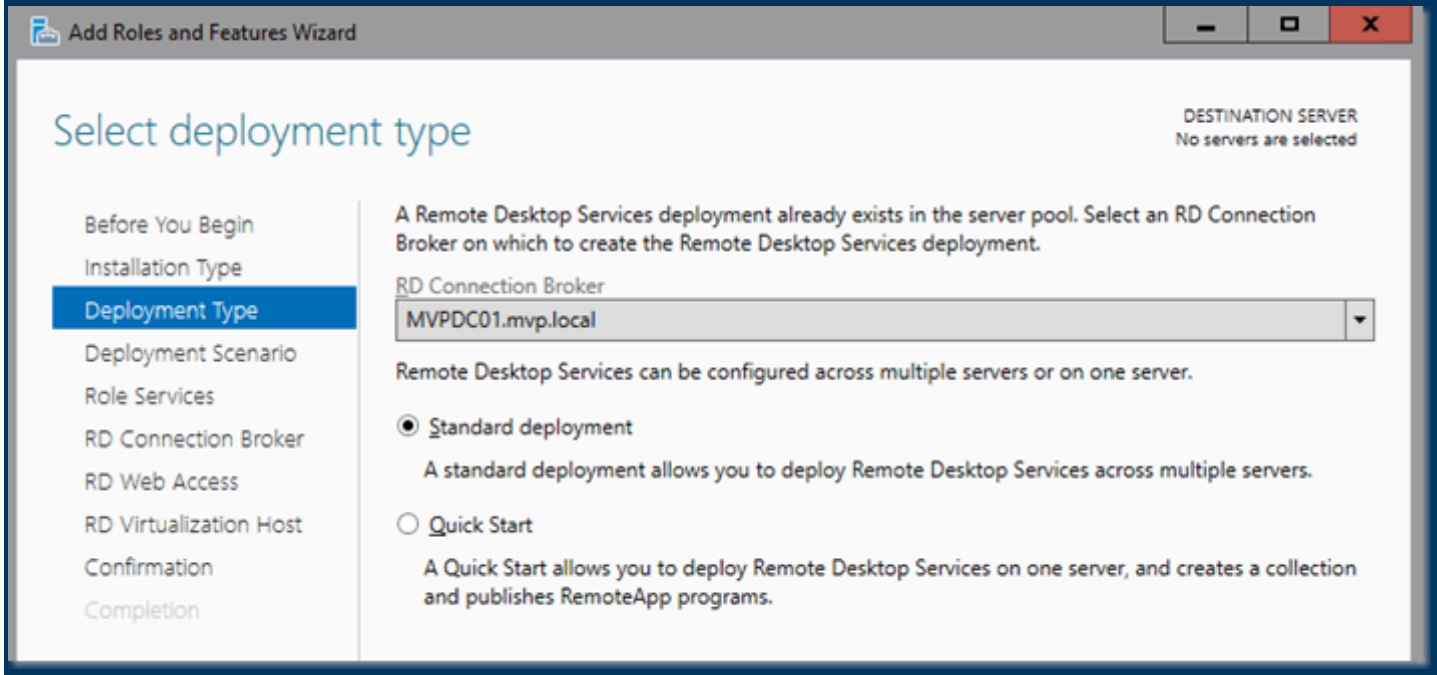


Selecting and installing the role. I did this in the menu but you can also do this in the configuration. and the role will be installed.



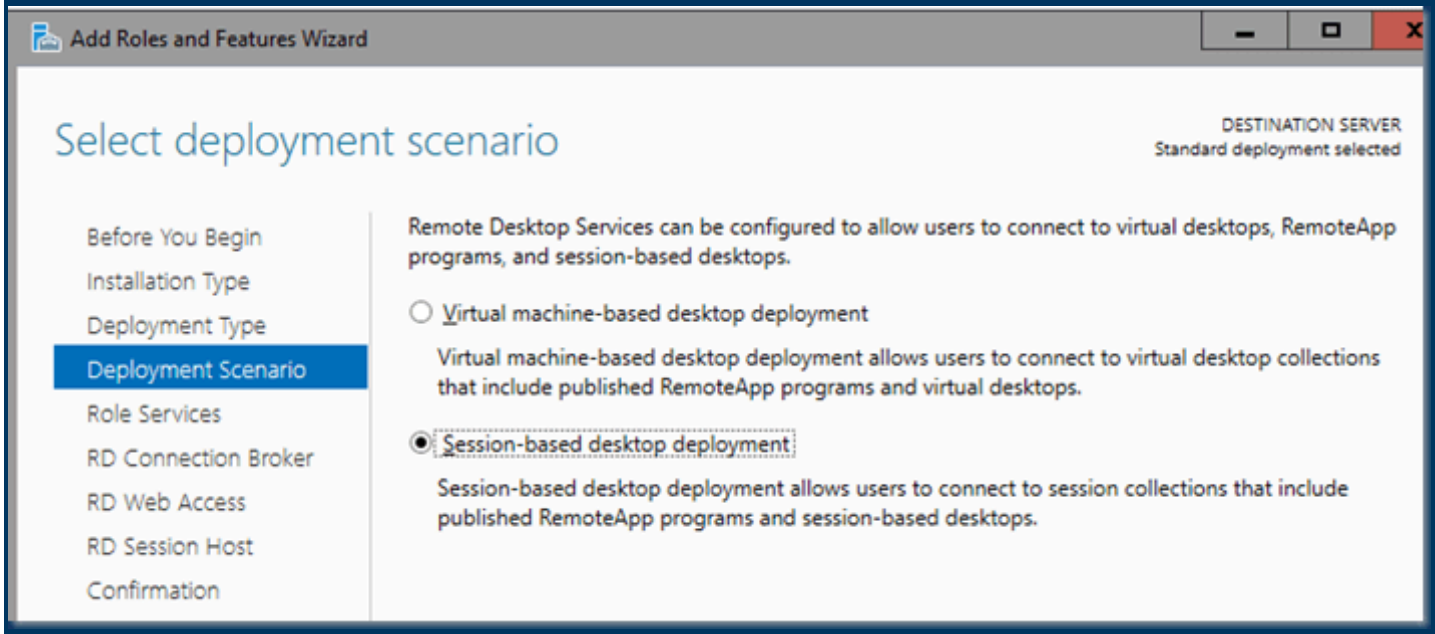
Now that the roles are installed there is an extra option in server manager <> Remote Desktop Services.

To configure Windows Server 2016 Remote Desktop Services you have to pick in the add roles and features the lower option Remote Desktop Services Installation.

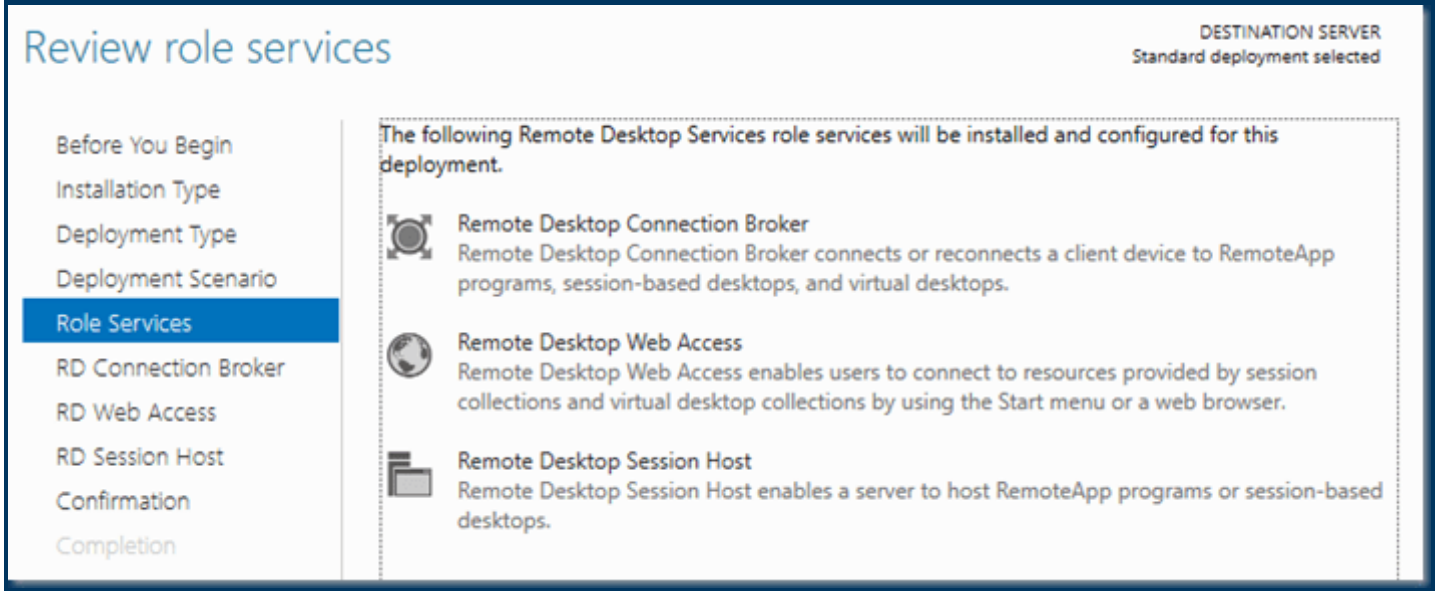


As you can see a quick Start option is here but we are not using this. and check the standard deployment. now you need to configure all the stuff.

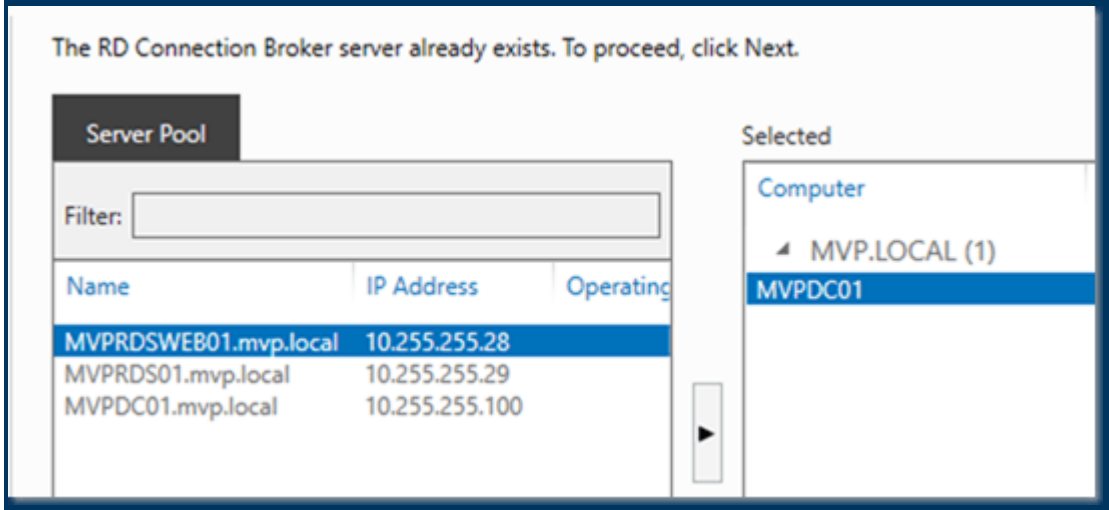
But for a quick demo you can pick the quick start option.



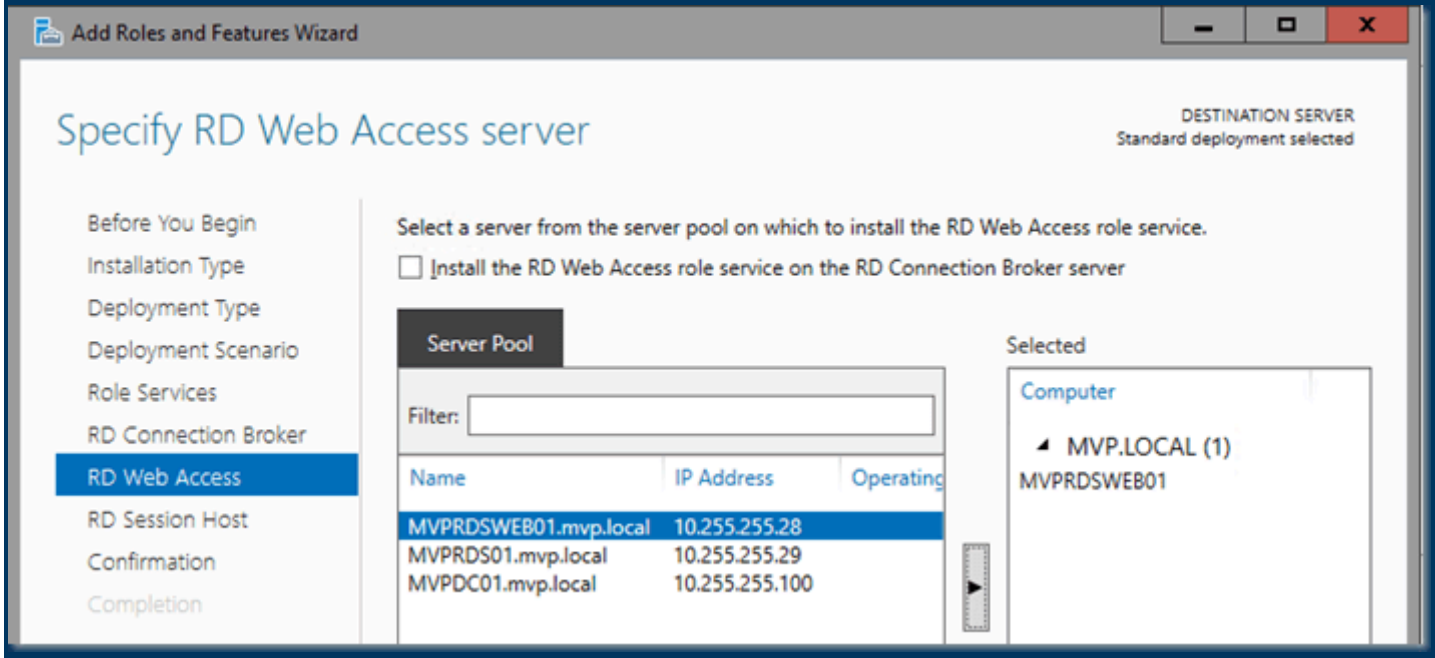
When using the VDI option you will need a machine that is running Hyper-v !. In my setup I'll use the Session based desktop deployment.



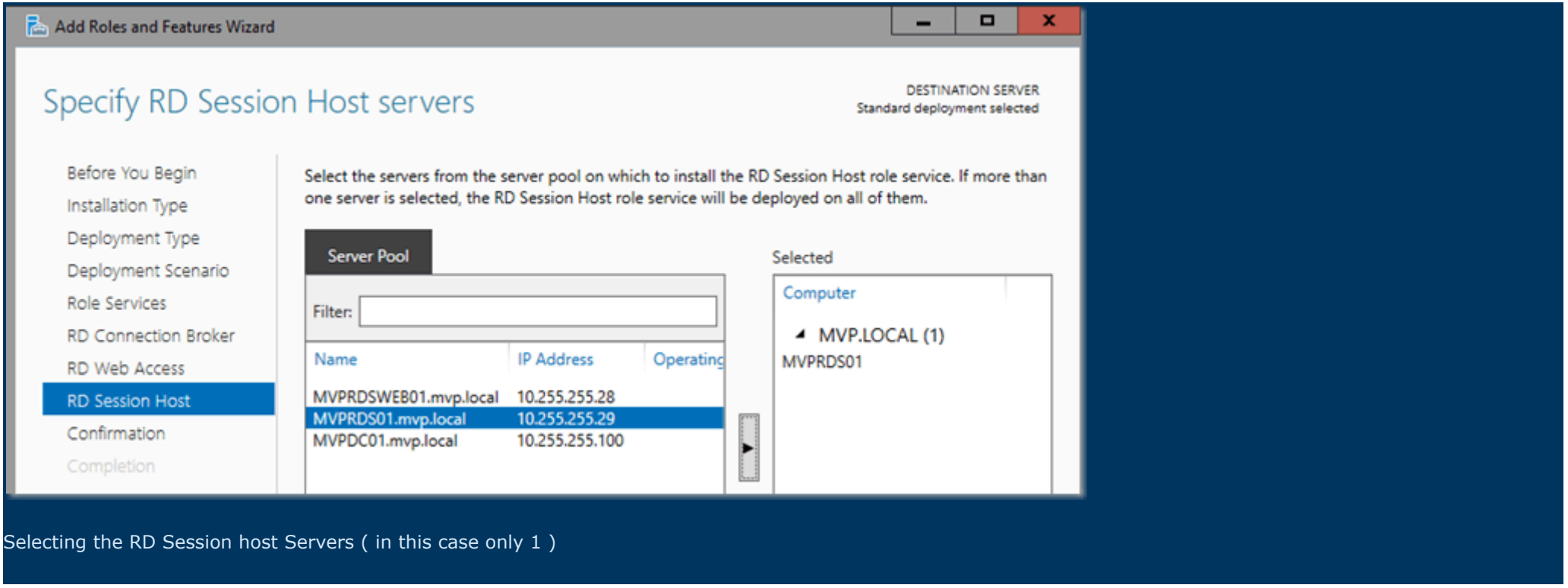
A quick overview of the roles that I'll need for this deployment.



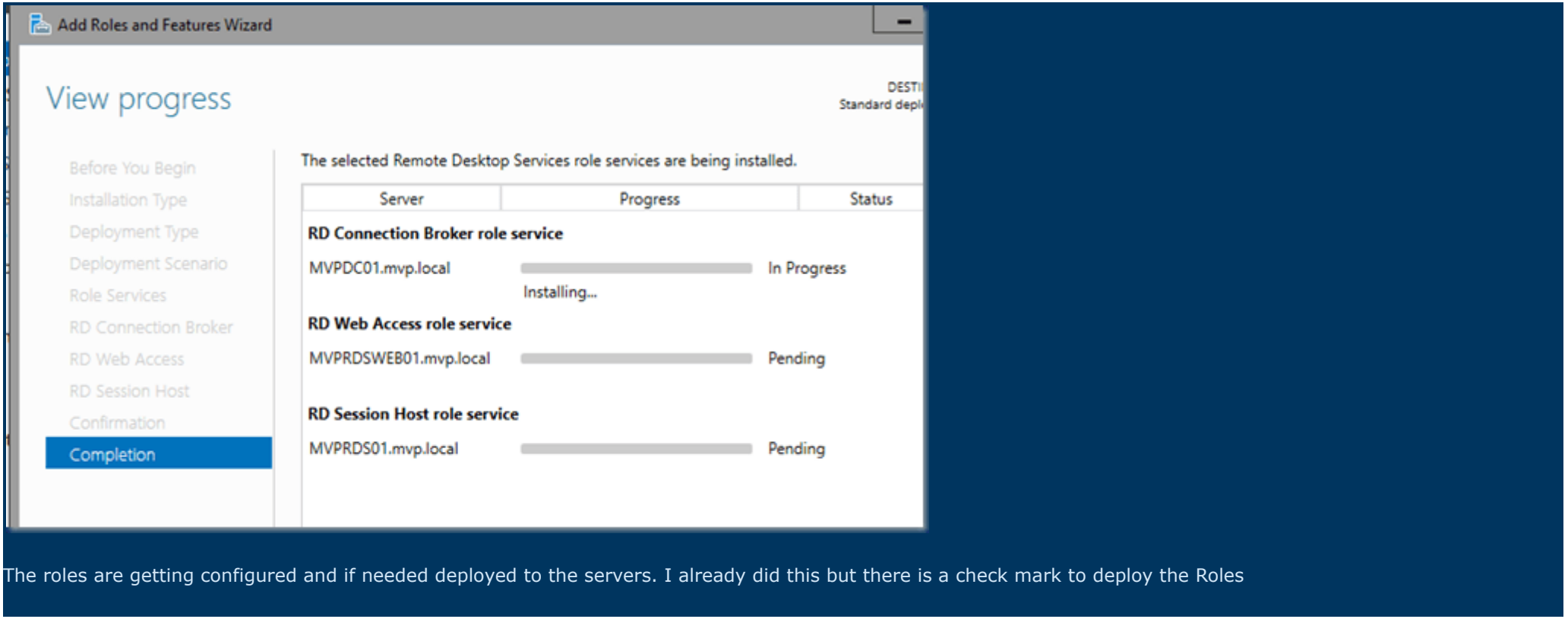
Selecting the RD Connection Broker Server



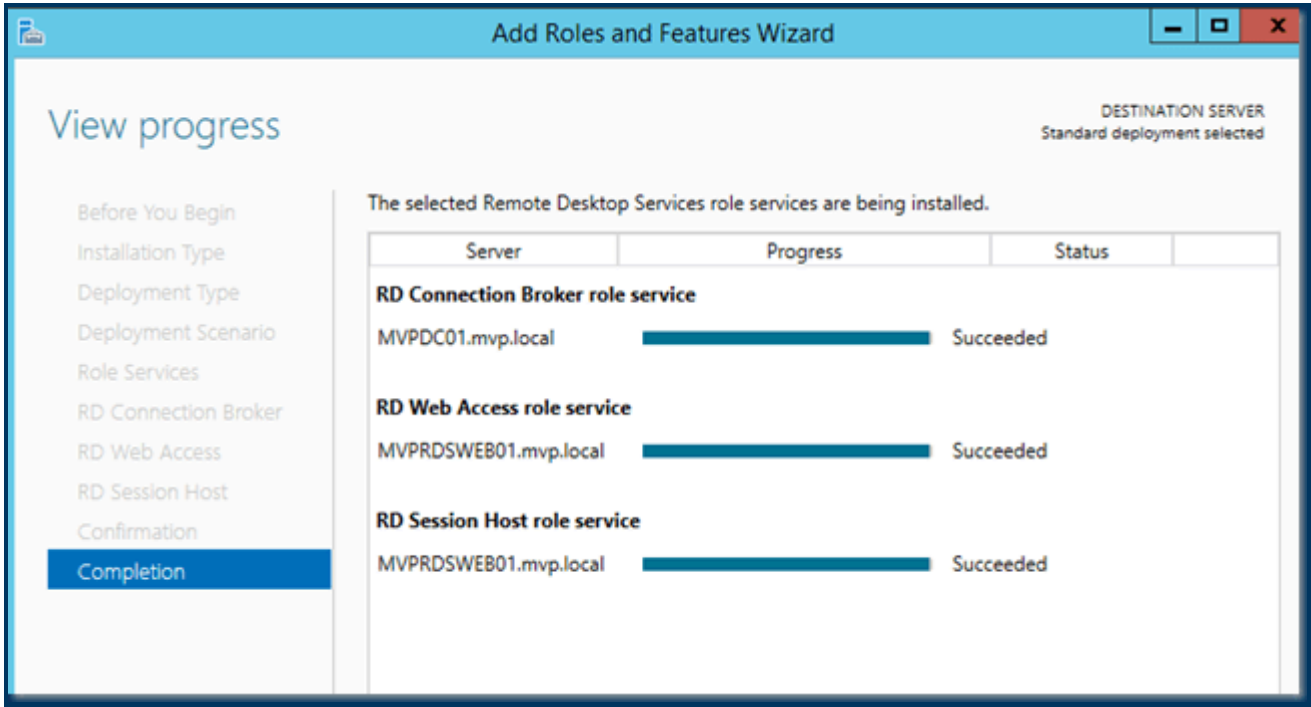
Selecting the RD Web Access Server



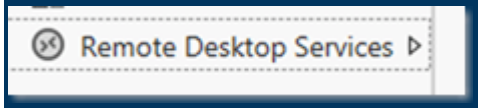
Selecting the RD Session host Servers (in this case only 1)



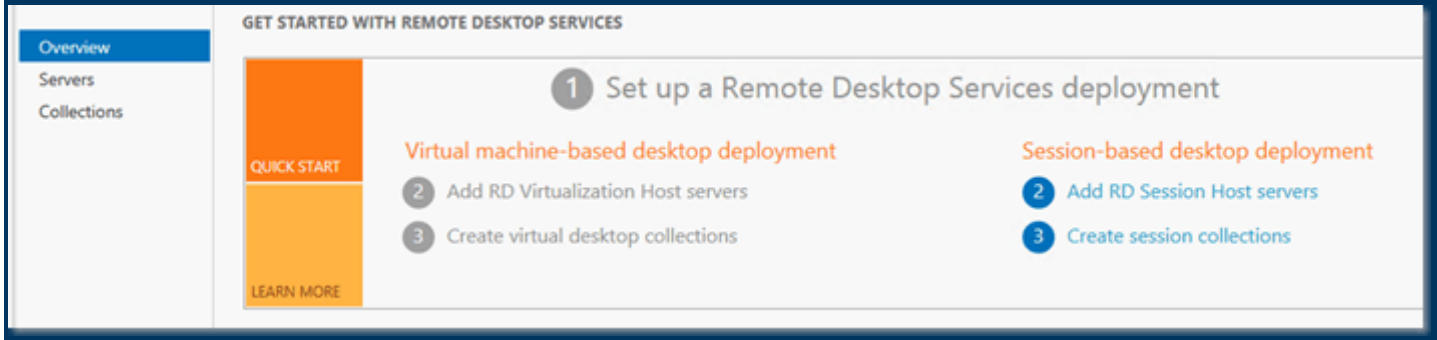
The roles are getting configured and if needed deployed to the servers. I already did this but there is a check mark to deploy the Roles

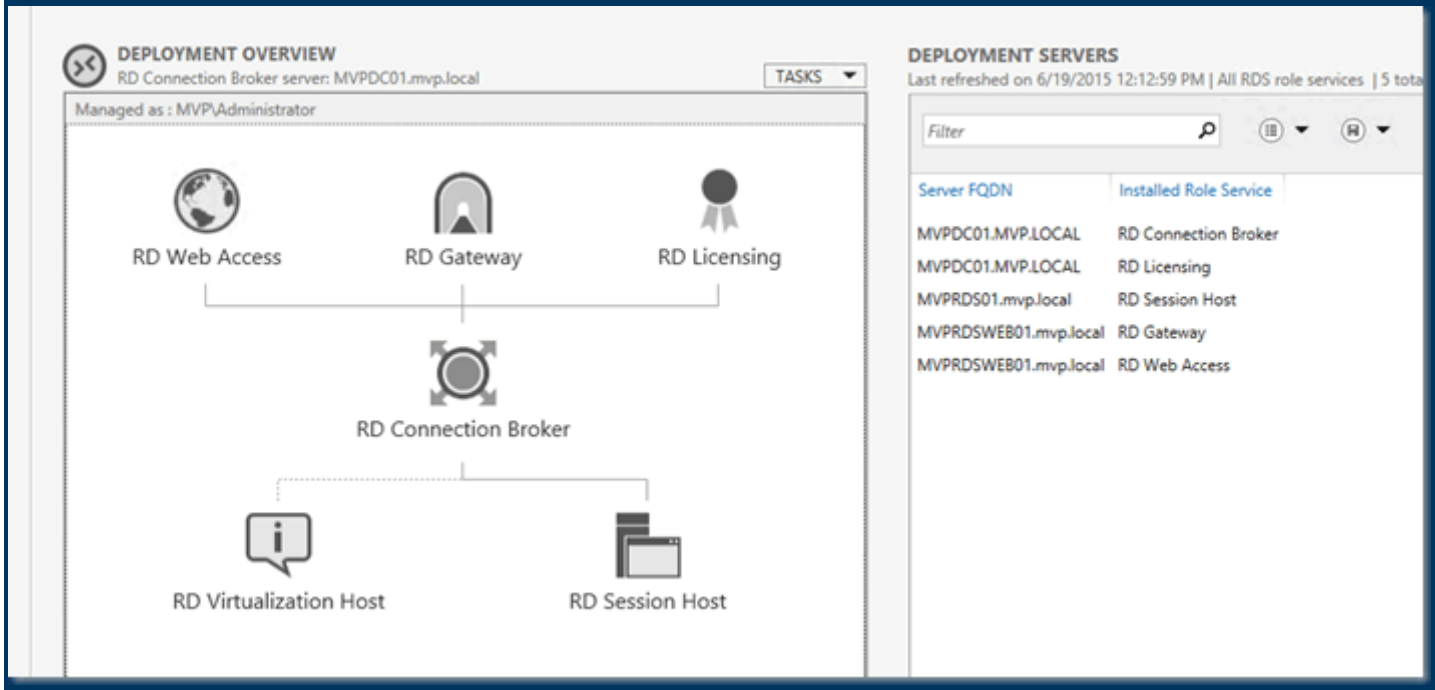


Now that all the roles are installed in server manager you can go to the Remote Desktop Services



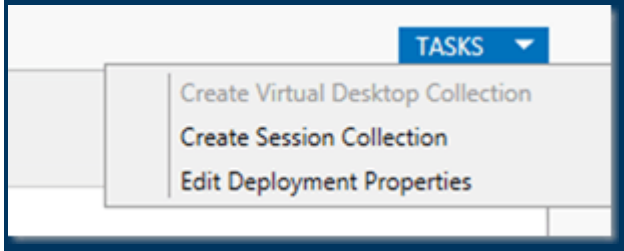
In the overview you can see what is deployed and what options you can do. but in every task pulldown item there are the same options.



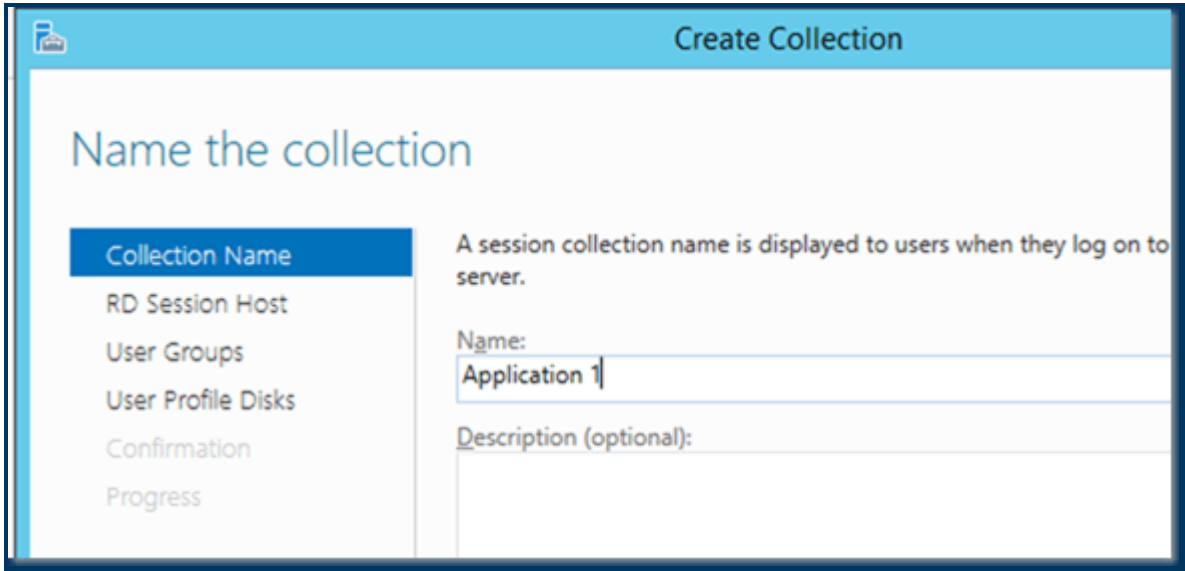


I installed all my options and I'm ready to create a Collection.

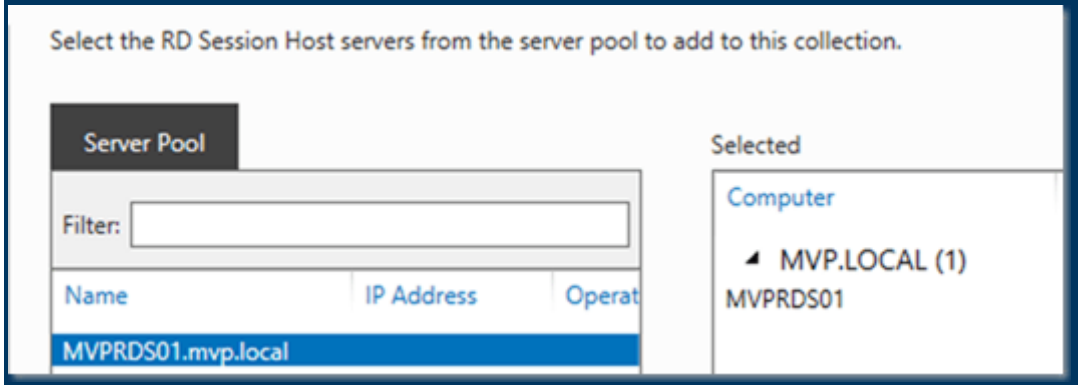
Create a Collection.



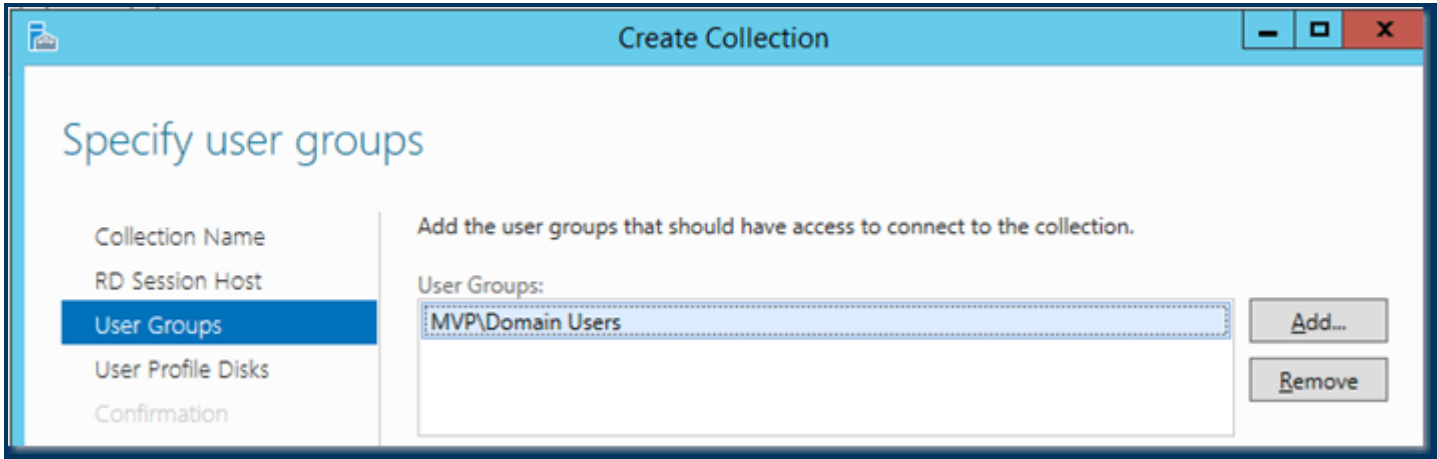
In the task menu I choose the Create Session Collection,



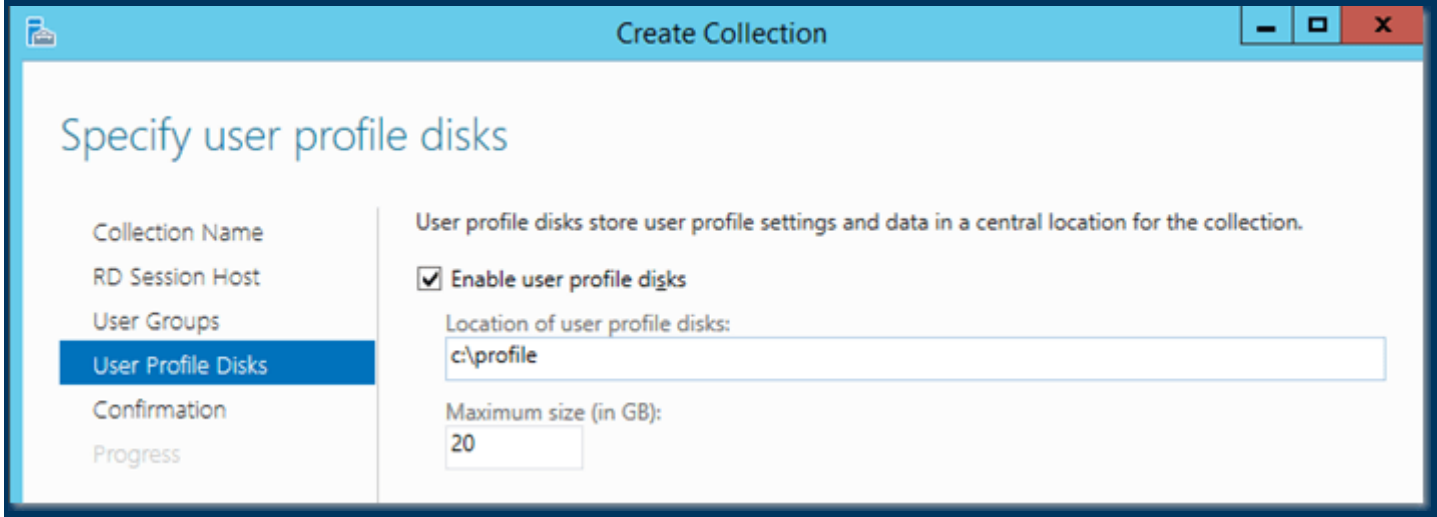
Just Name it



Choose a RD Session host Servers



What users may access this collection. I'll pick all domain users.



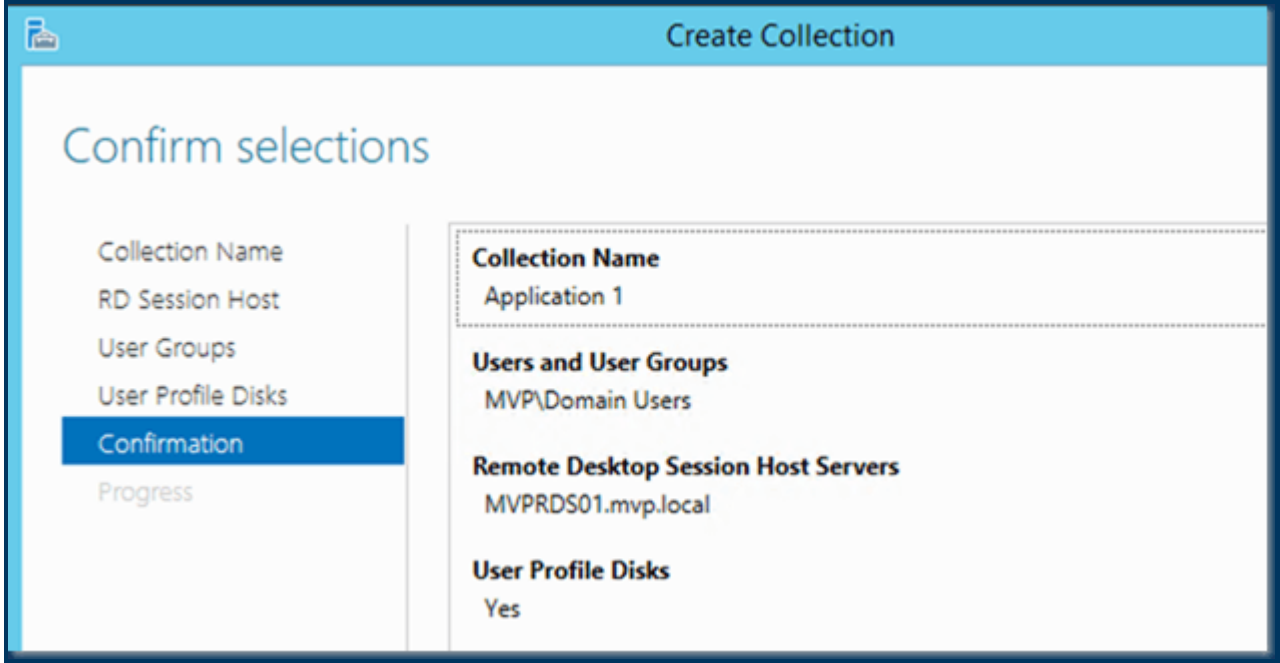
User profile disks offer several advantages:

- Configuration and deployment is simpler than roaming profiles or folder redirection.
- User profiles can be maintained even on pooled virtual desktops that get rolled back after logoff.
- Logon and logoff times are reduced.

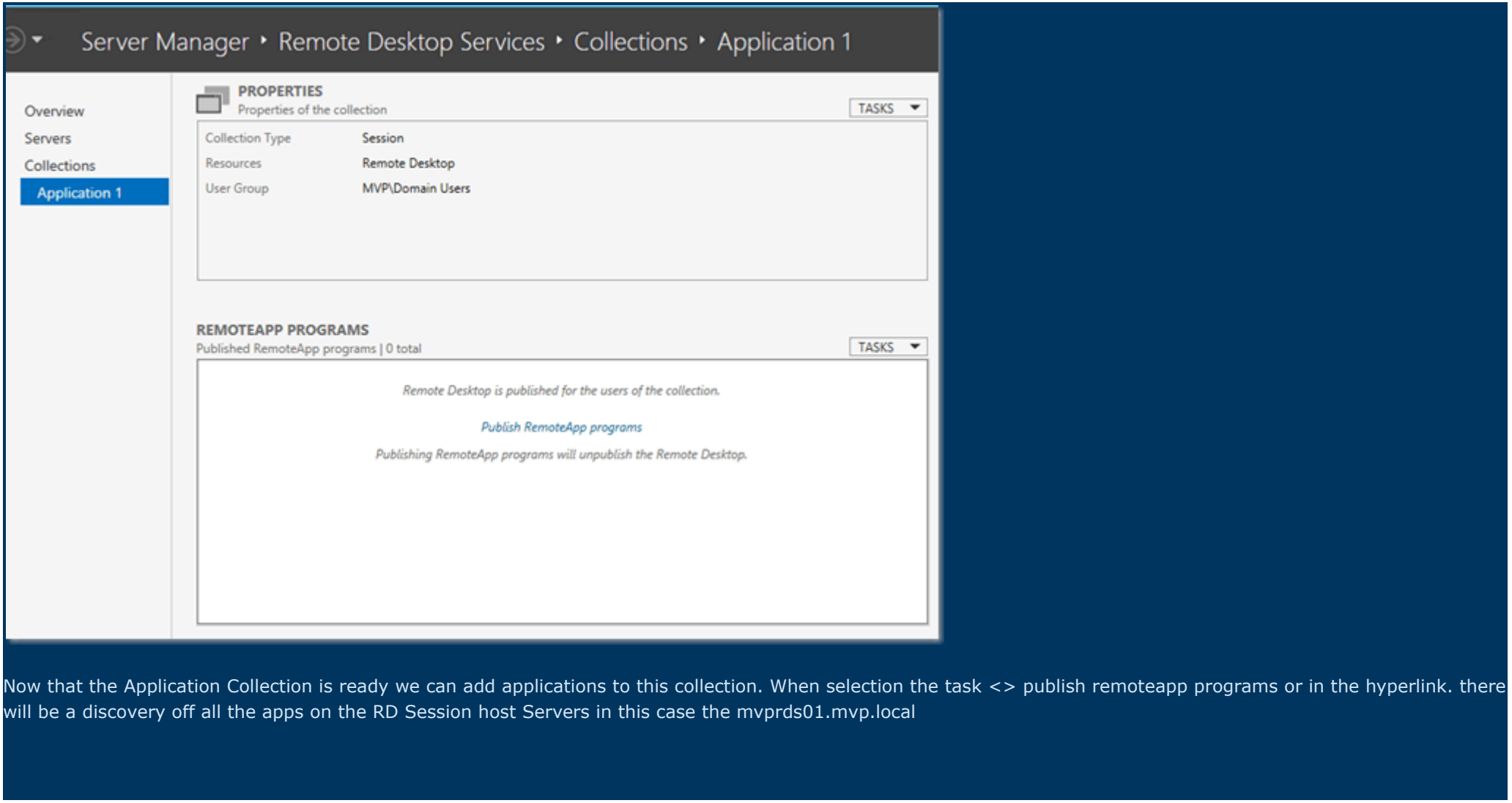
- Previously, profiles could be corrupted if used simultaneously on multiple computers. User profile disks are specific to the collection, so they can't be used on multiple computers simultaneously.
- Administrators can have granular control of exactly which locations get saved to the virtual hard disk (VHDX).
- User profile disks can be stored on Server Message Block (SMB) shares, cluster shared volumes, SANs, or local storage.
- In pooled virtual desktop collections, user profile disks work with virtual machines running both Windows 8 and Windows 7 with Service Pack 1 (SP1).

Some things to remember about user profile disks:

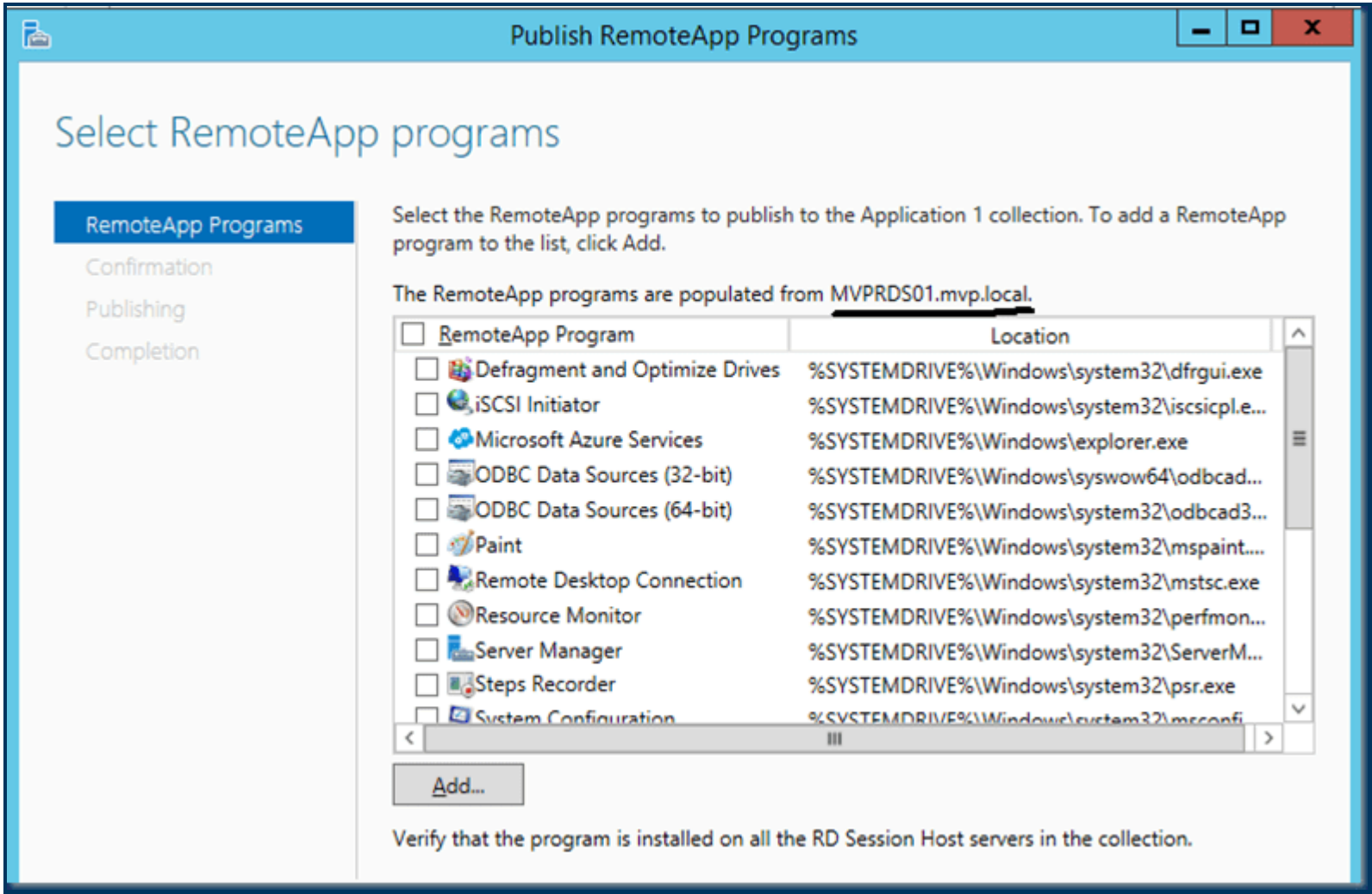
- User profile disks are available only in pooled virtual desktop collections and session collections—not in personal virtual desktop collections.
- Share permissions are automatically set up by the management tools.
- Use Server Manager or Windows PowerShell to manage user profile disks.
- User profile disks are for a single collection only. A user connecting to two different collections will have two separate profiles. If you want to synchronize settings, refer to Microsoft User Experience Virtualization.



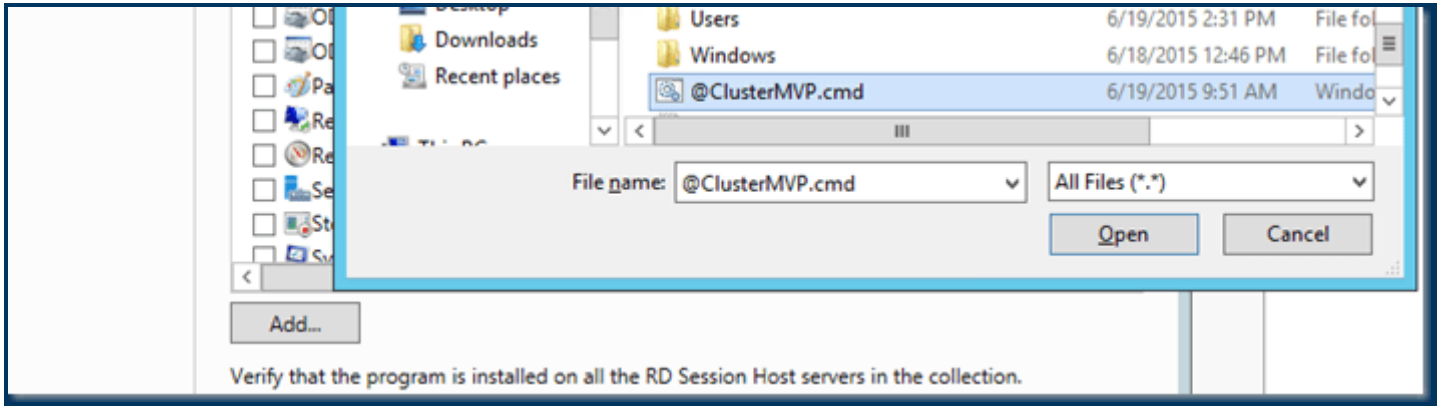
When Creating the collection we can make a start for publishing applications.



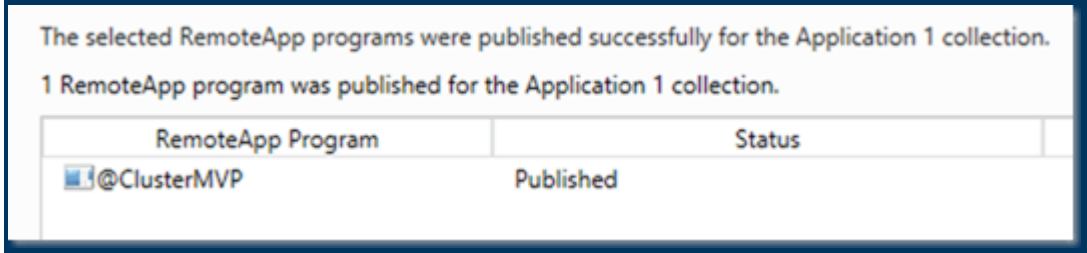
Now that the Application Collection is ready we can add applications to this collection. When selection the task <> publish remoteapp programs or in the hyperlink. there will be a discovery off all the apps on the RD Session host Servers in this case the mvprds01.mvp.local



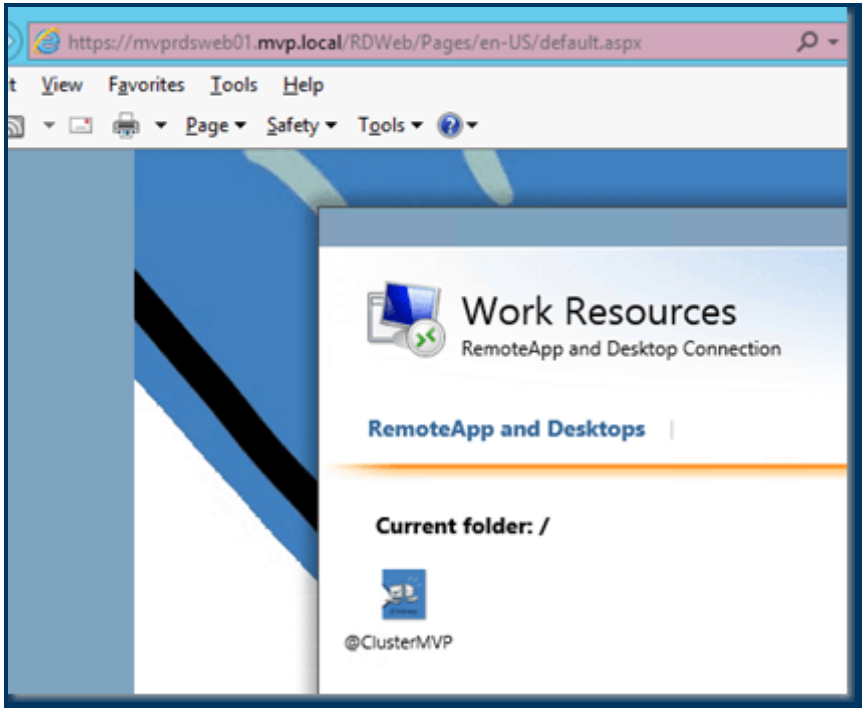
But sure you can apps that are not discovered just press add



and press Publish and there is the APP

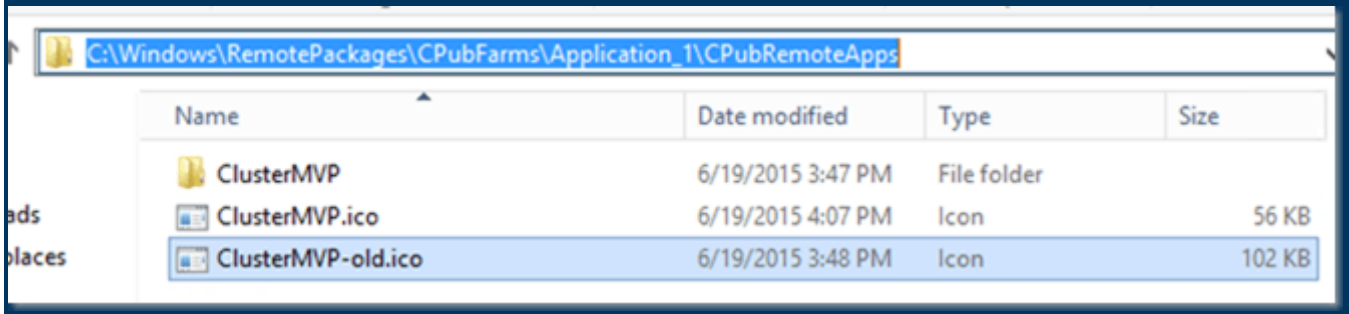


When Logon to the Portal you can see the RemoteApp



Changing the Icon of the RemoteApp can be done by PowerShell or copy and replace. On the RDS Broker server. goto the path :

C:\Windows\RemotePackages\CPubFarms\Application_1\CPubRemoteApps



all the RemoteApps are there and can be changed here.

OR change the ICON with the shell23.dll with powershell

To change the Icon

The Icon Index for this interface works top to bottom, starting with 0. So count the rows until you see your desired icon, multiply this by 4, subtract 1, and count up to your desired icon. The Icon Index for the Windows Update icon turns out to be 46.

Type one of the following commands in the Powershell box:

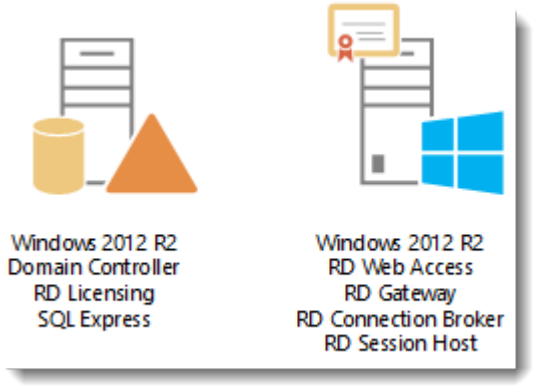
```
Get-RDRemoteApp -Alias "clustermvp" | Set-RDRemoteApp -IconPath "c:\windows\system32\shell32.dll" -IconIndex 46
```

Step by Step Windows 2012 R2 Remote Desktop Services –[mySqL]

A step by step guide to build a Windows 2012 R2 Remote Desktop Services deployment.

Part 1 – Deploying a single server solution.

Although it is called a single server installation, we will need 2 servers as shown below.



Software used in this guide:

Windows Server 2012 R2 ISO (evaluation can be downloaded here:http://technet.microsoft.com/en-us/evalcenter/dn205286.aspx)

SQL Server 2012 SP1 Express x64 With tools (free version can be downloaded here:http://www.microsoft.com/en-us/download/details.aspx?id=35579. After clicking the download button select SQLEXPRWT_x64_ENU.exe)

SQL Server 2012 SP1 Native Client (free version can be downloaded here:http://www.microsoft.com/en-us/download/details.aspx?id=35580. After clicking the download button select ENU\x64\sqlncli.msi)

And a certificate. I got mine for free from https://startssl.com. This certificate needs to contain the FQDN you will use as the RD Web Access URL (mine is gateway.it-worxx.nl in this guide). It needs to be in .pfx format and you need to have the private key in it.

This guide will not focus on building a domain using a single domain controller and adding the second server as a member server to this domain.

Also some basic knowledge is assumed in this guide. I will not detail how to create a Security Group and adding a computer account to it. I will also not detail how to install SQL Express, or adding logins to a SQL Server Instance security context. If you need extra help with this, Bing it or drop me a mail with details, and I will provide steps to continue.

I will be using Hyper-V 3.0 on my Windows 8.1 laptop and I have prepared 2 servers:

ITWDC01 (1 vCPU, 512MB memory, dynamic, 60GB Harddisk)

Installed Windows IPv4 192.168.66.20/24

Added .NET Framework 3.5 as a feature

Added Active Directory Domain Services as a role

Configured this server as a Domain Controller in a new forest: itw.test

ITWRDS01 (1 vCPU, 512MB memory, dynamic, 60GB Harddisk)

Installed Windows

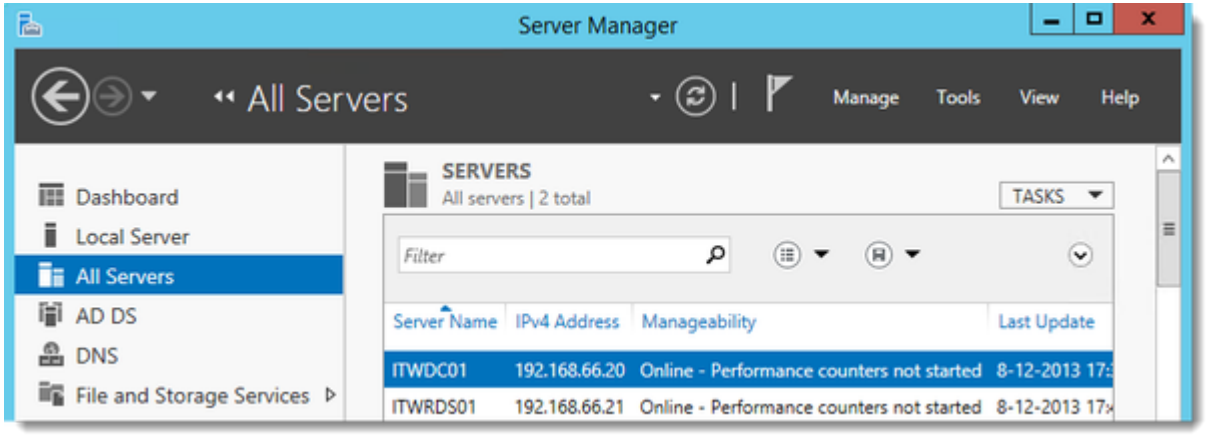
Added .NET Framework 3.5 as a feature

IPv4 192.168.66.21/24, DNS server 192.168.66.20

Configured it as a member server in the itw.test domain

Installing the Remote Desktop Services Roles

Log on to the Domain Controller, and in Server Manager right-click the All Servers node and add the second server using the Add Servers command (or select the All Servers node, click Manage and click Add Servers).

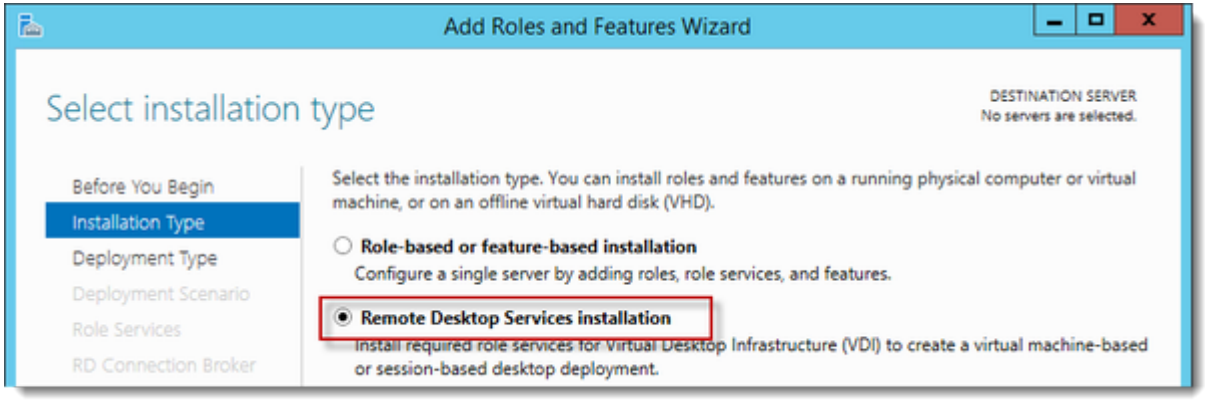


Now that all servers needed in this deployment scenario are present, click Manage, and click Add Roles & Features.

Before you begin

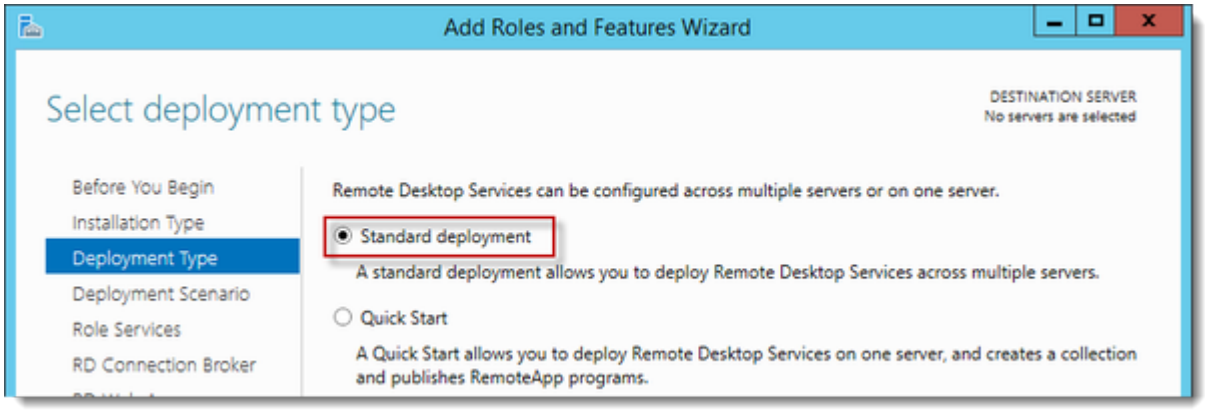
Click Next.

Select Installation Type



Select Remote Desktop Services installation. Click Next.

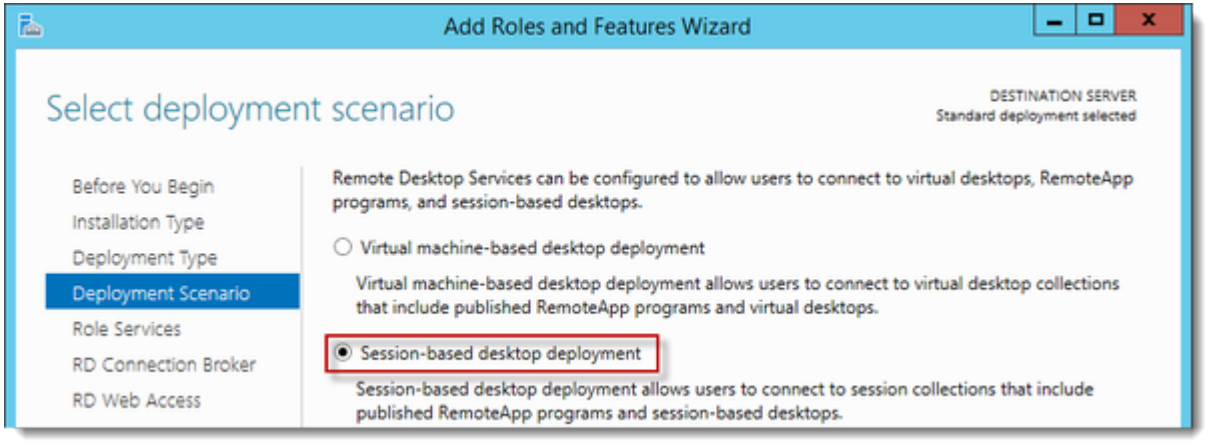
Select Deployment Type



Although Quick Start might be a valid option for a single server deployment, leave the default selected. This will explain the steps necessary to install Remote Desktop Services in greater detail.

Click Next.

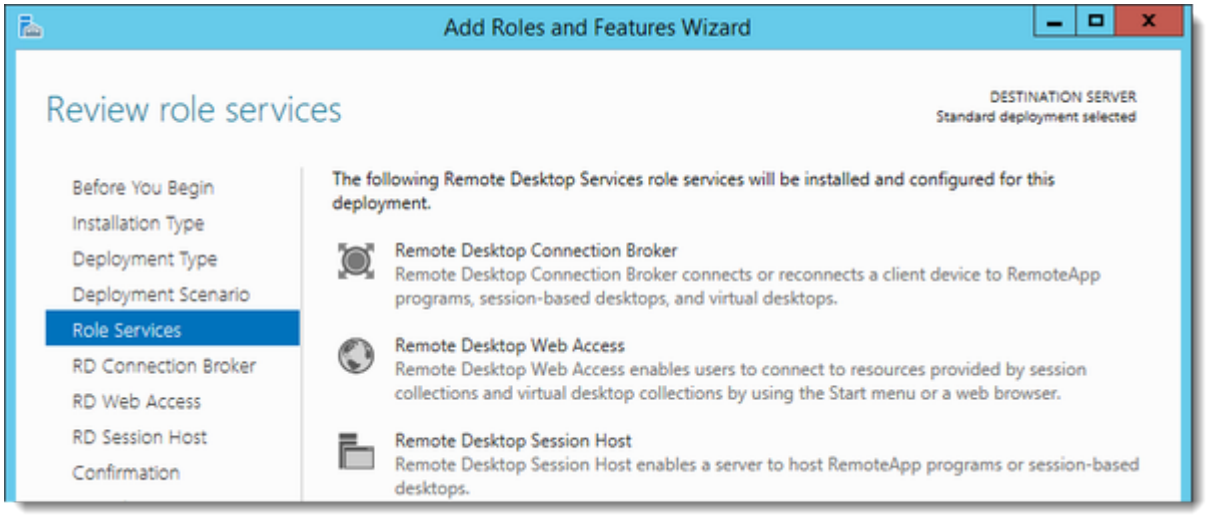
Select Deployment Scenario



Select Session-based desktop deployment. The other option will be a different post in this series.

Click Next.

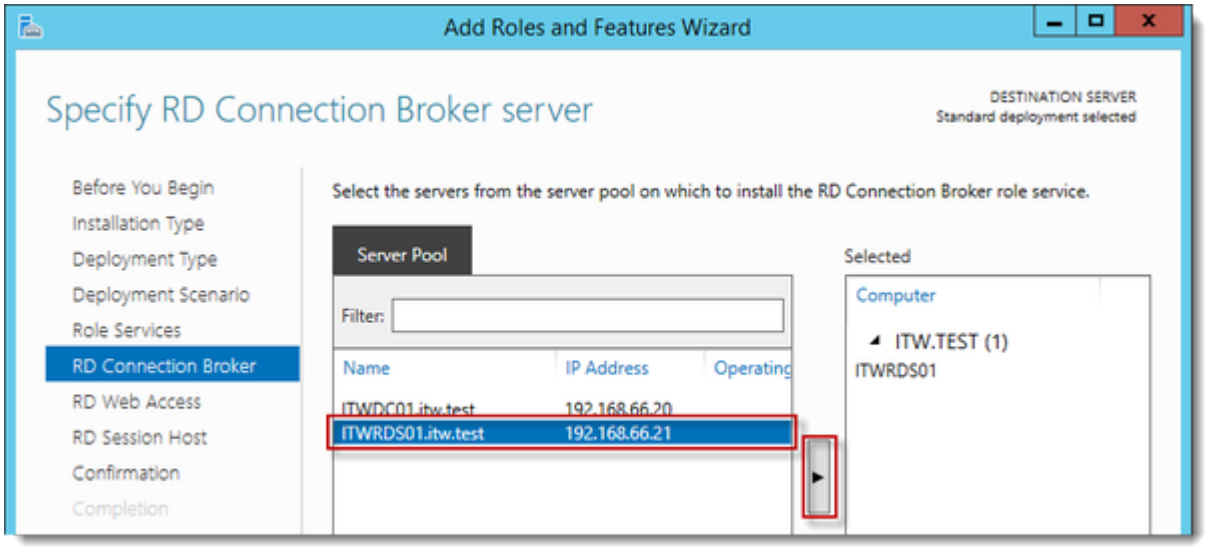
Review Role Services



Review the services that will be installed.

Click Next.

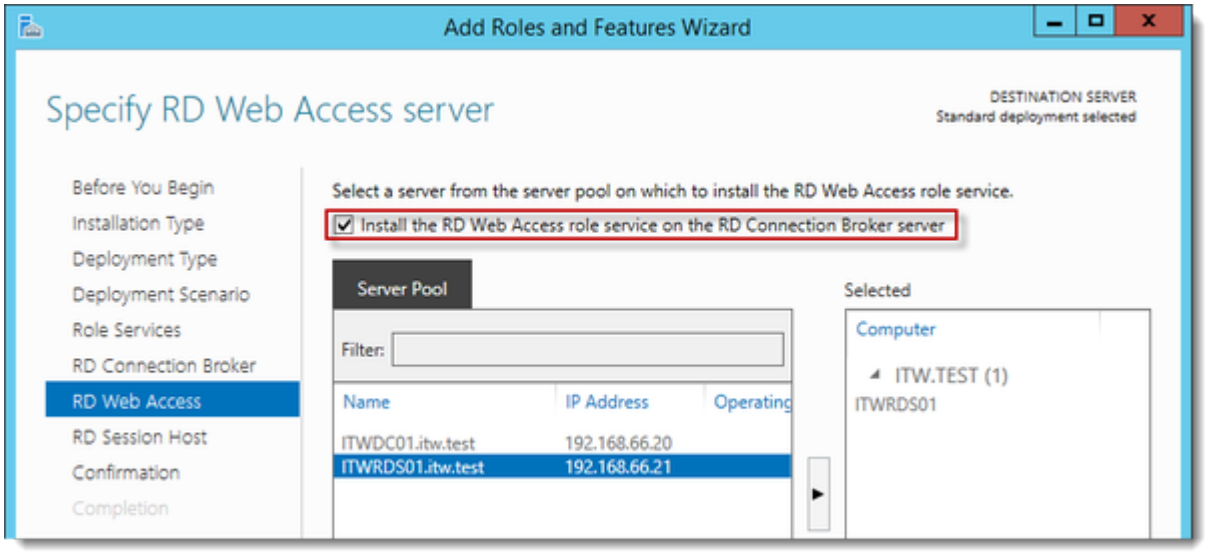
Specify RD Connection Broker server



Click the member server and click the Add button.

Click Next.

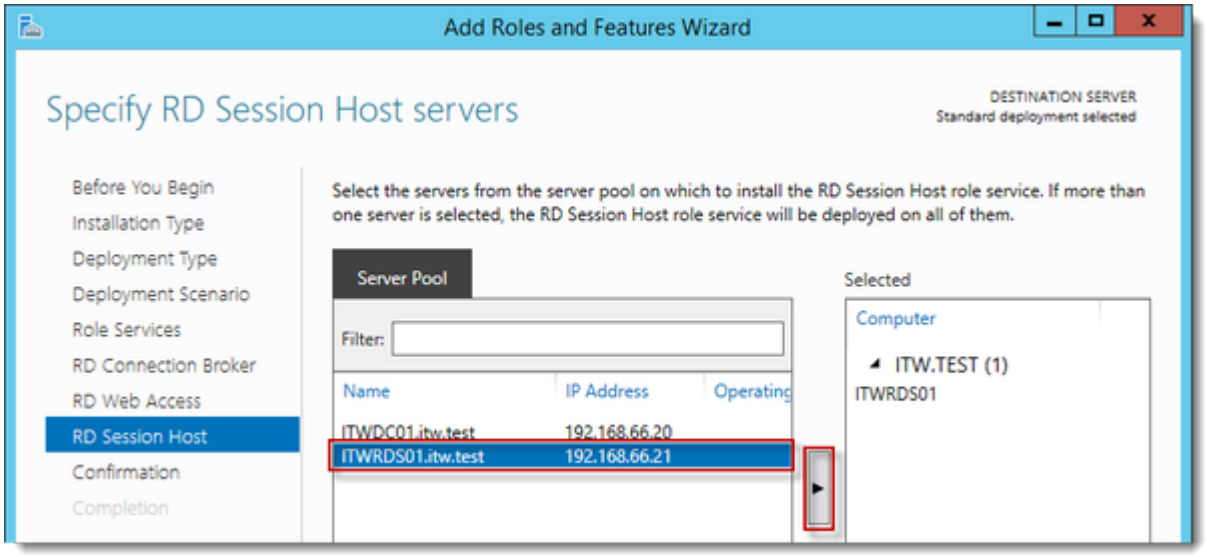
Specify RD Web Access server



Check Install the RD Web Access role on the RD Connection Broker server.

Click Next.

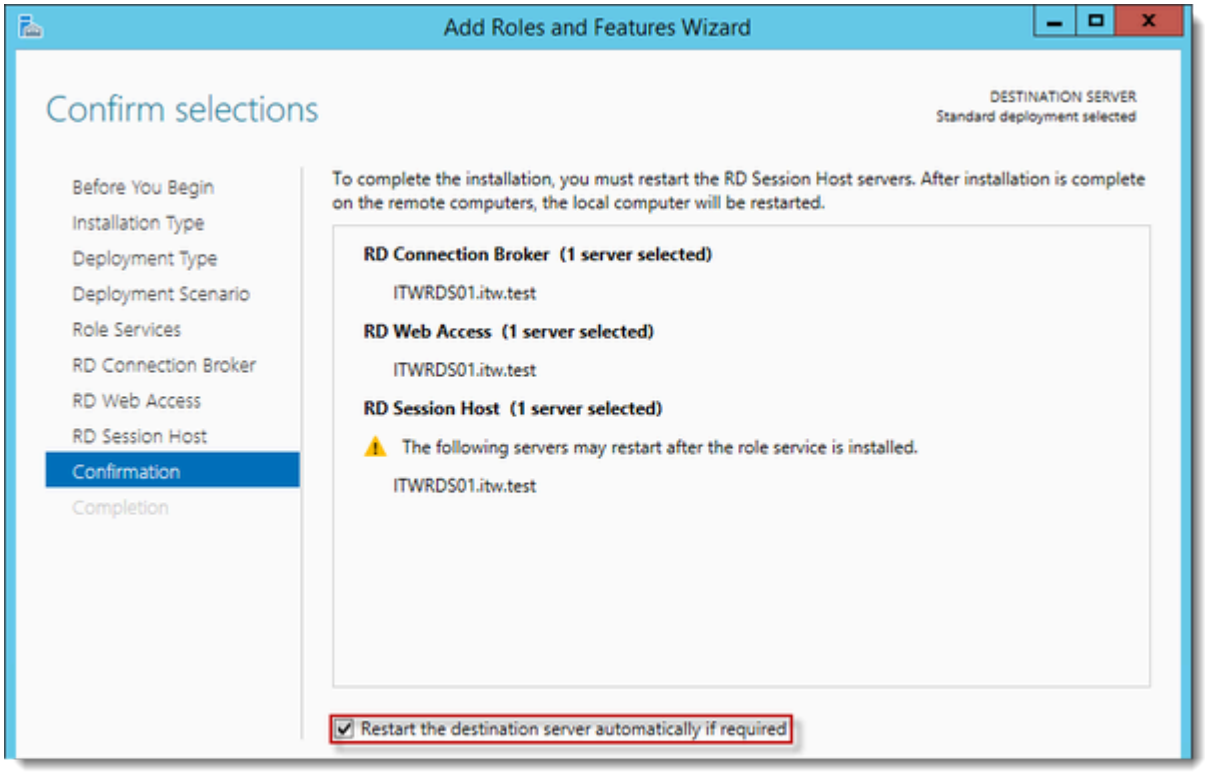
Specify RD Session Host server



Click the member server and click the Add button.

Click Next.

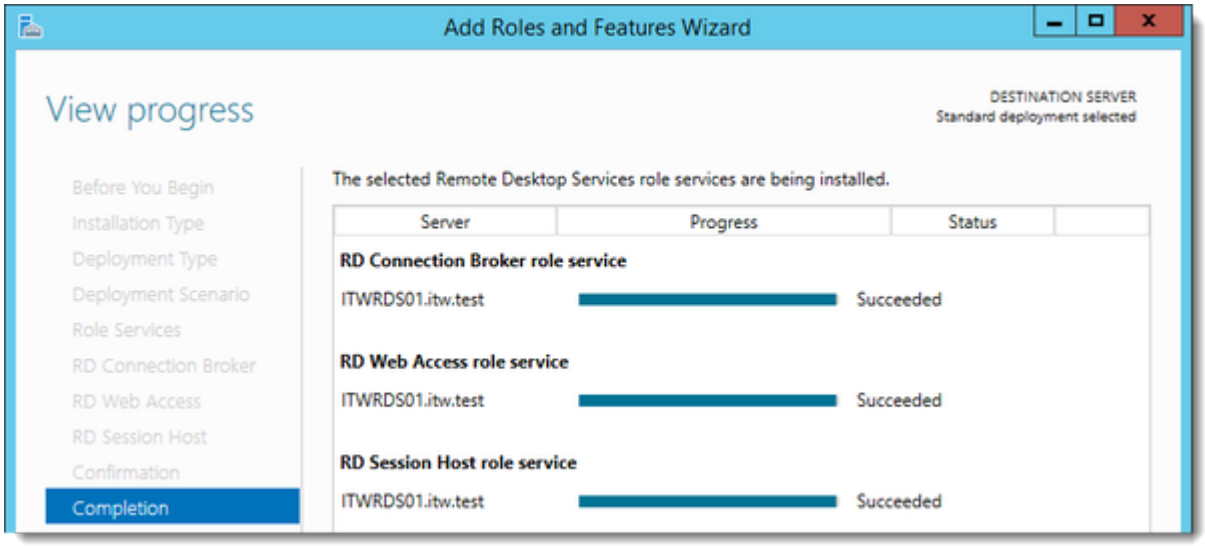
Confirm selections



Check Restart the destination server automatically if required.

Click Deploy.

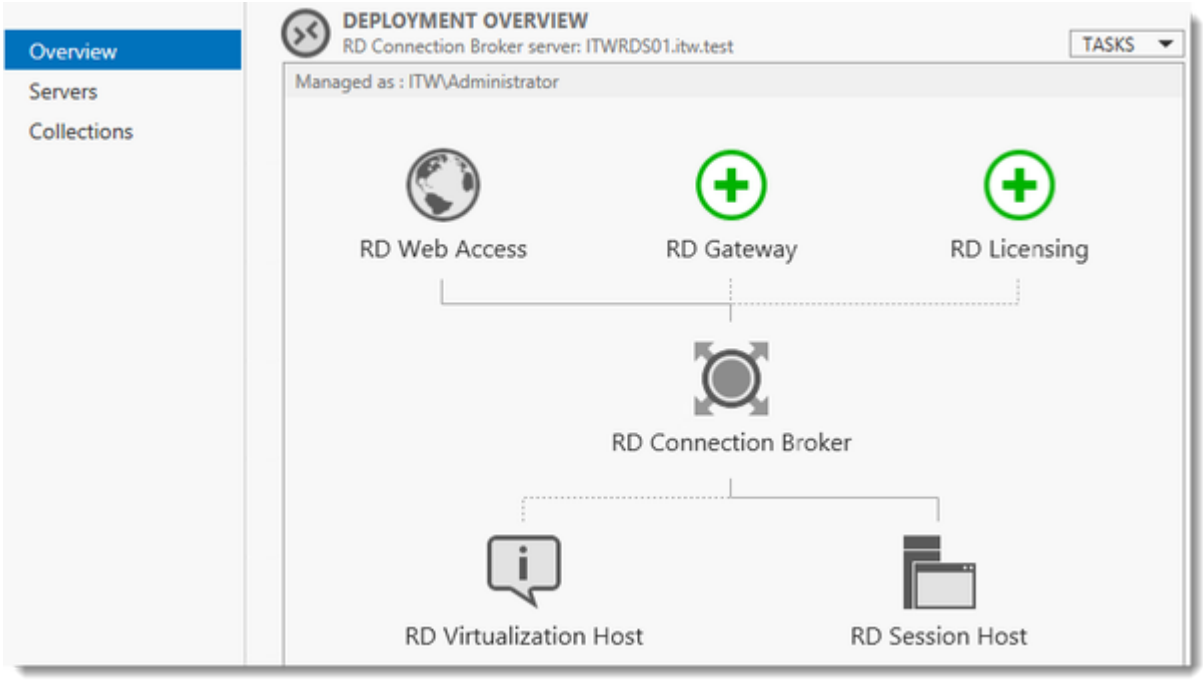
View progress



Wait until all role services are deployed and the member server has restarted.

Click Close.

In Server Manager click Remote Desktop Services and scroll down to the overview.

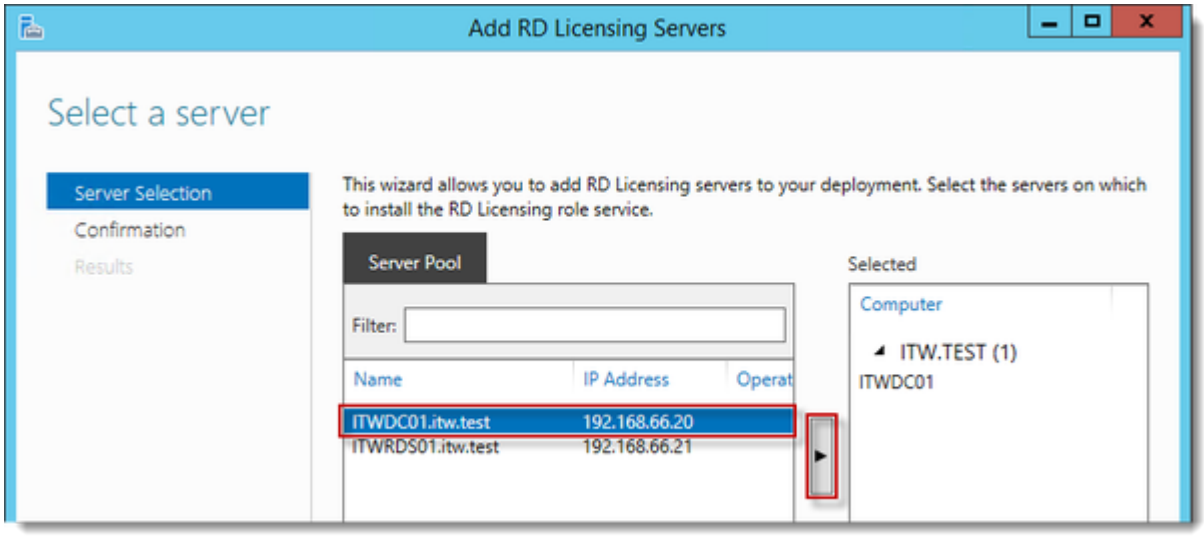


As you can see the deployment is missing a RD Gateway server and a RD Licensing server.

Installing the missing Remote Desktop Services Roles

Click the Add RD Licensing server button.

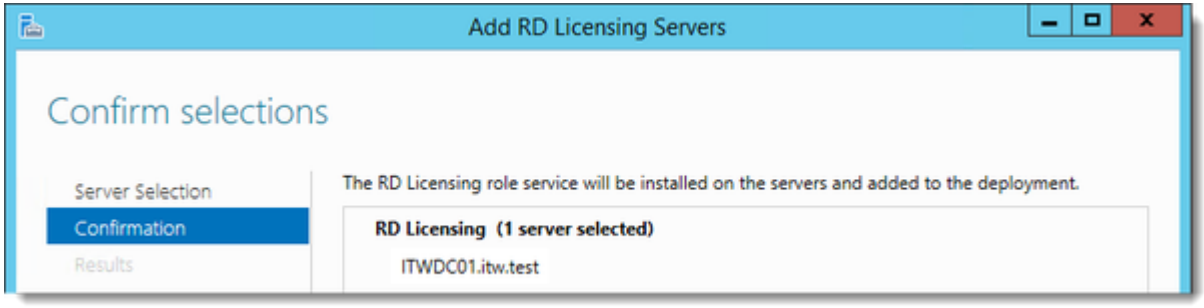
Select a server



Click the domain controller and click the Add button.

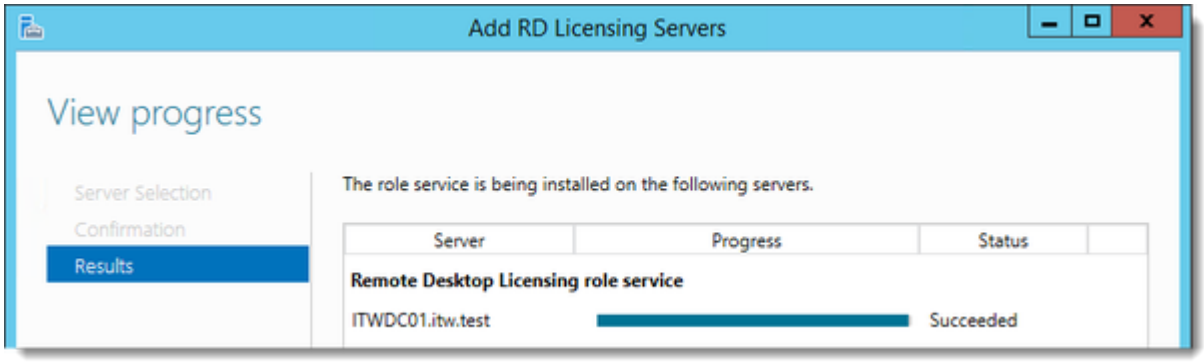
Click Next.

Confirm selections



Click Add.

View progress

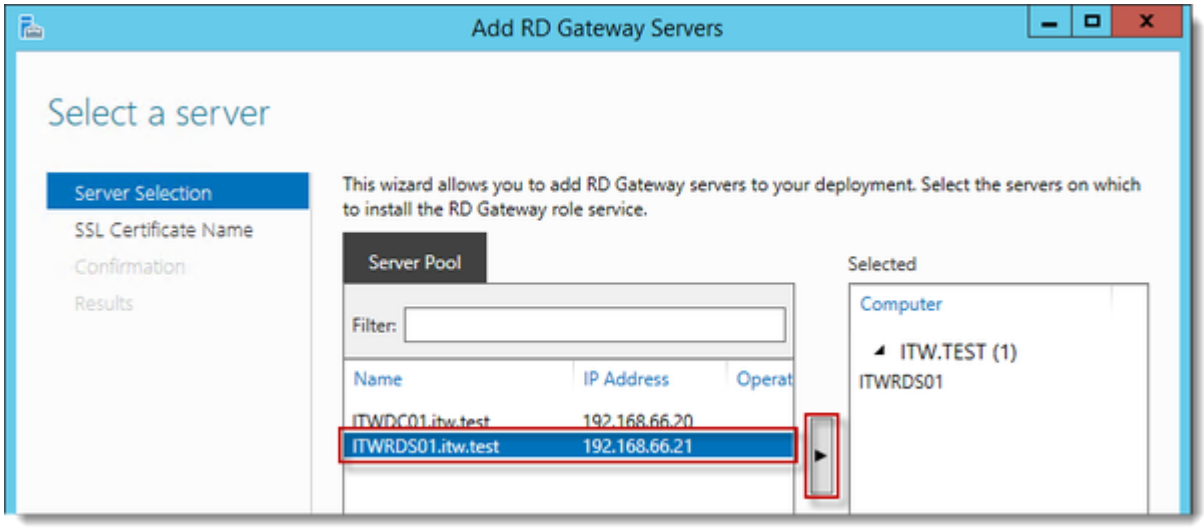


Wait until the role service is deployed. No restart is needed.

Click Close.

Click the Add RD Gateway server button.

Select a server



Click the member server and click the Add button.

Click Next.

Name the self-signed SSL certificate

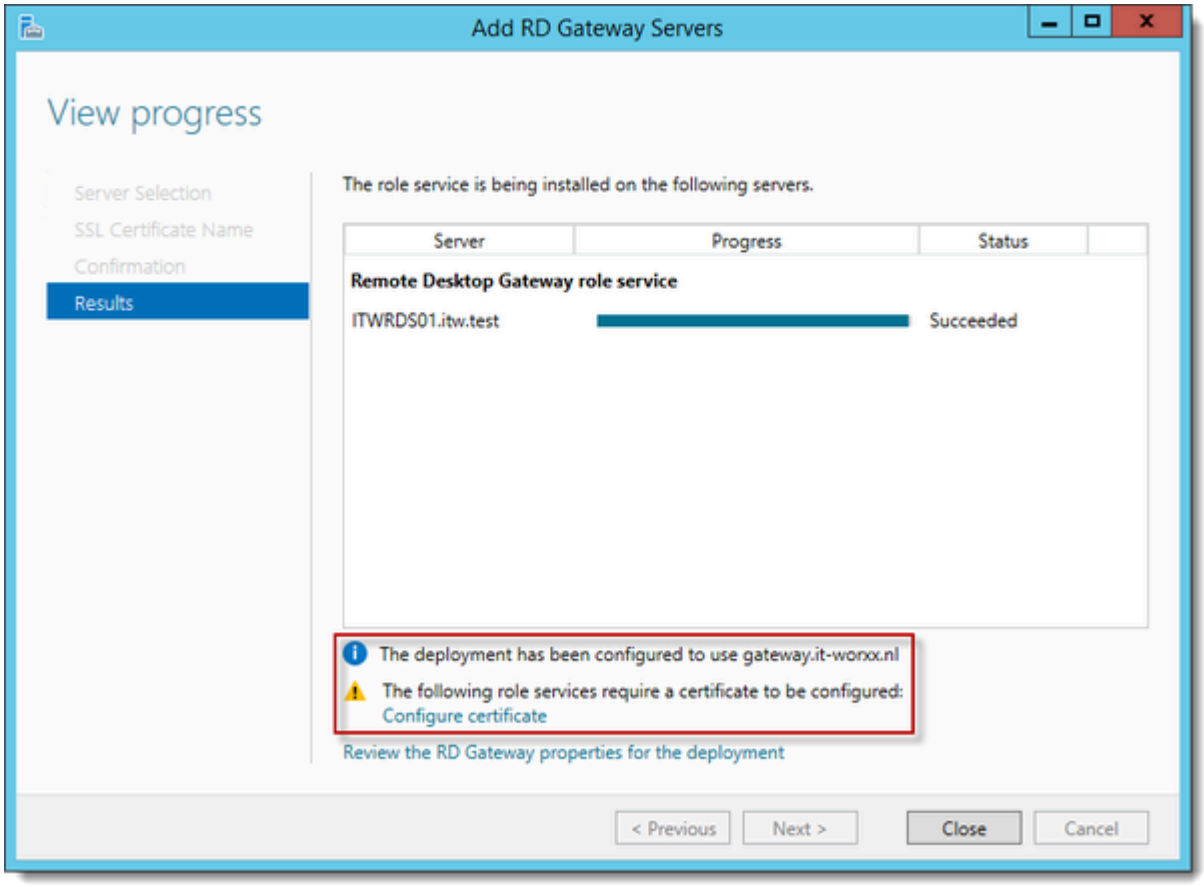
The wizard creates a self-signed certificate. We will deal with certificates in this deployment in a little bit. Enter the external Fully Qualified Domain Name which you will also use for the Web Access URL. In my case, for lack of a better name, I used “gateway.it-worxx.nl”. I didn’t want to use “remote.it-worxx.nl” or “desktop.it-worxx.nl” or anything else.

Click Next.

Confirm selections

Click Add.

View progress



Wait until the role service is deployed. No restart is needed.

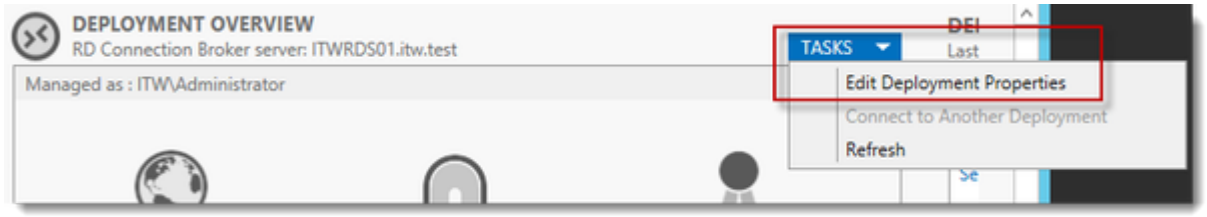
Notice that “gateway.it-worxx.nl” was configured for the deployment.

Also notice that even more certificate configuring is need, but we’ll get to that later. Pay no attention to it for now.

Click Close.

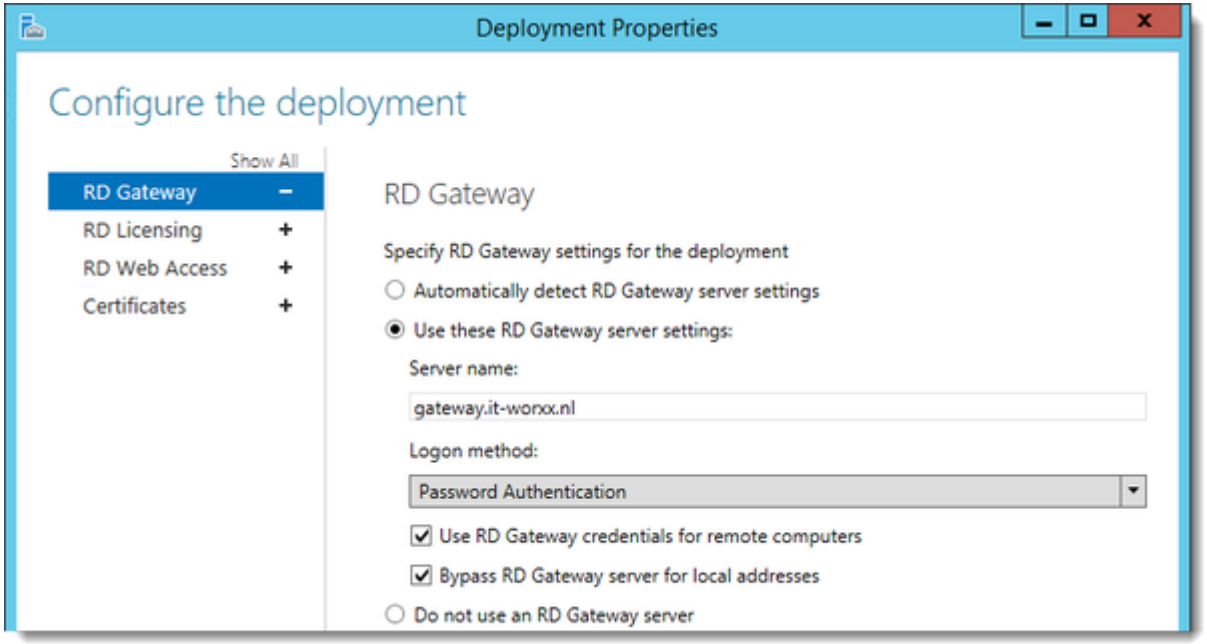
Let’s have a quick look at the certificate configuration.

Reviewing the Remote Desktop Services certificate requirements



In Server Manager, Remote Desktop Services, Overview, click Tasks and click Edit Deployment Properties.

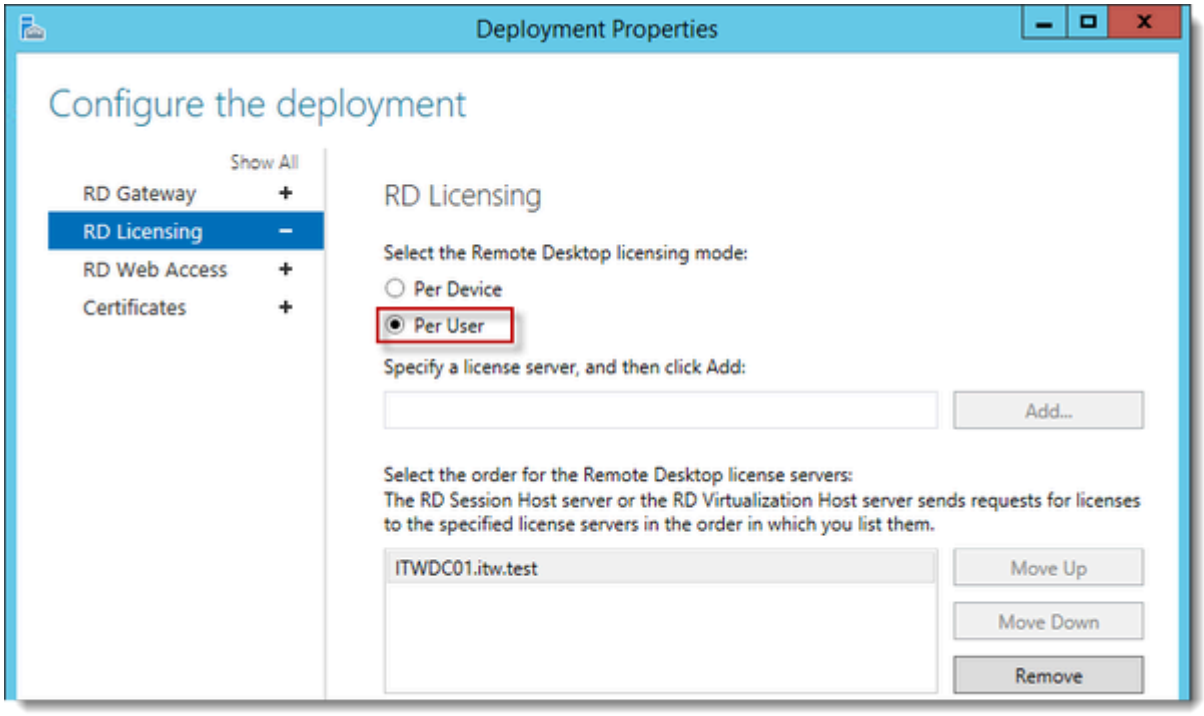
Configure the deployment



Review the RD Gateway settings and notice what settings are available.

Click RD Licensing.

Configure the deployment

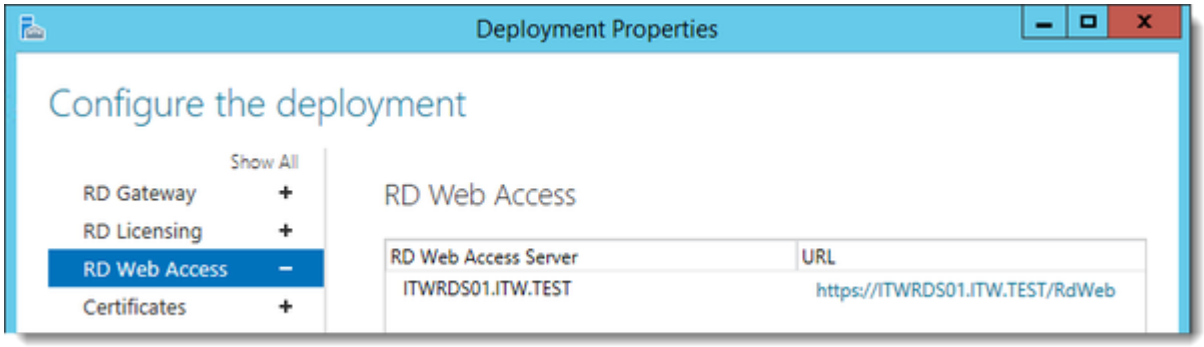


Notice that a RD License server is available, but no license type is selected yet.

I selected Per User, but since this is just a guide setup, it really doesn’t matter.

Click RD Web Access.

Configure the deployment

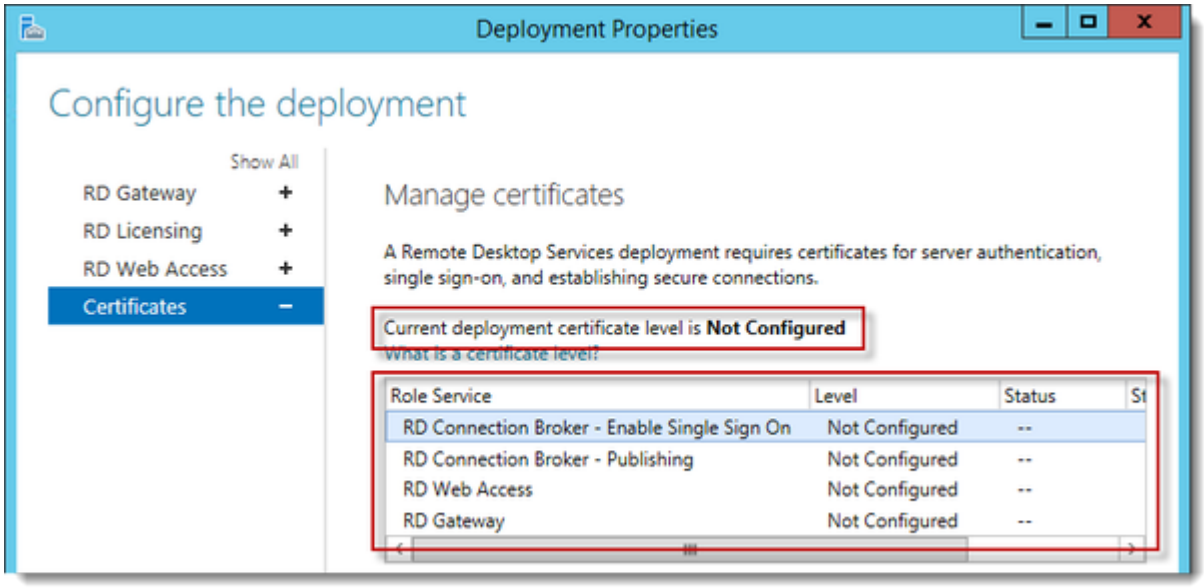


By default the RD Web Access IIS application is installed in /RdWeb. If you want to know how to change this, check another post:

https://msfreaks.wordpress.com/2013/12/07/redirect-to-the-remote-web-access-pages-rdweb/

Click Certificates.

Configure the deployment



Notice that the certificate level currently has a status of Not Configured.

As you can see, certificates are used for different goals within the deployment.

The RD Gateway certificate is used for Client to gateway communication and needs to be trusted by the clients. Either install the self-signed certificate on all clients, or use a certificate for which the complete certificate chain is already trusted by all clients. As it said in the wizard, the external FQDN should be on the certificate.

The RD Web Access certificate is used by IIS to provide a server identity to the browser clients (and to the Feed clients, but that's a subject for a future post).

The RD Connection Broker actually has two goals for which it needs certificates. To enable single sign on (server to server authentication), and for publishing (signing RDP files). If you look in the deployment you'll see that the Connection Broker is now configured to use "itwrds01.itw.test", so we have to change it to use an external FQDN as well.

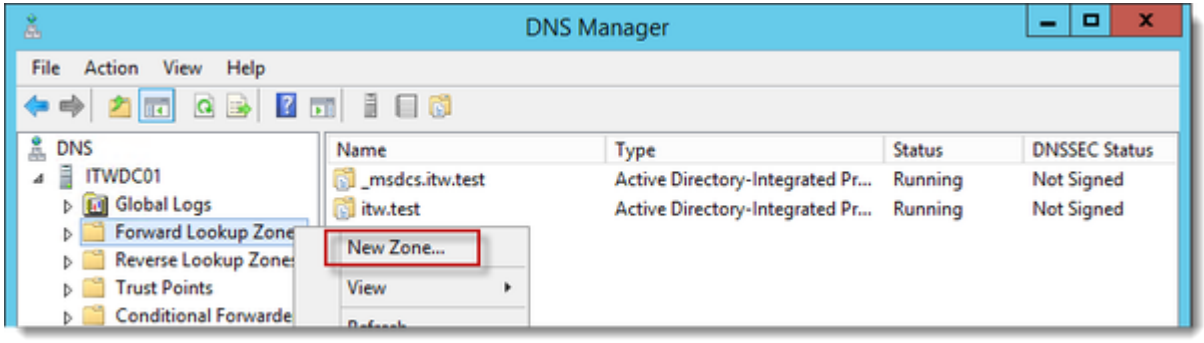
If we use the same FQDN for all goals described above, we need only 1 certificate, and only 1 external IP address.

We'll come back to this wizard later to assign the certificate. First order of business is to change the internal FQDN for the Connection Broker to an external FQDN.

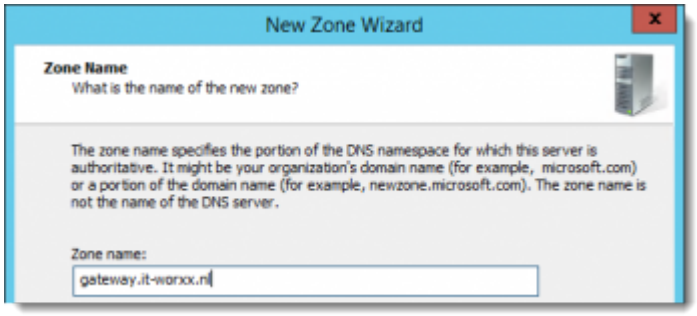
Click OK (no reason why we shouldn't commit the change we made on the licensing tab, remember?)

Preparing for completing the Remote Desktop Services configuration

Open DNS Manager on the domain controller and browse to Forward Lookup Zones.



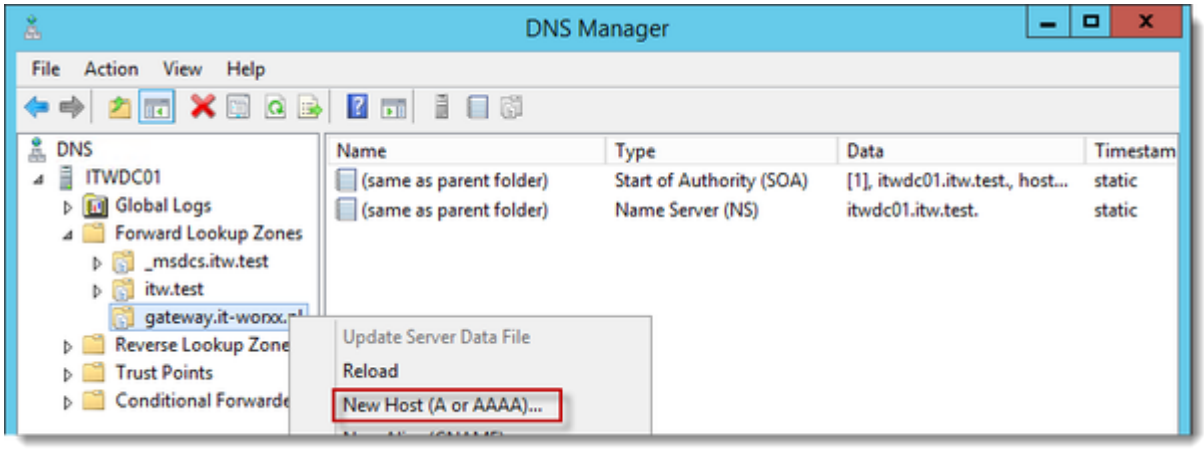
Right click Forward Lookup Zones and click New Zone… Go through this wizard accepting the defaults until you have to enter a Zone Name.



Enter the external FQDN which will also be used by the Connection Broker.

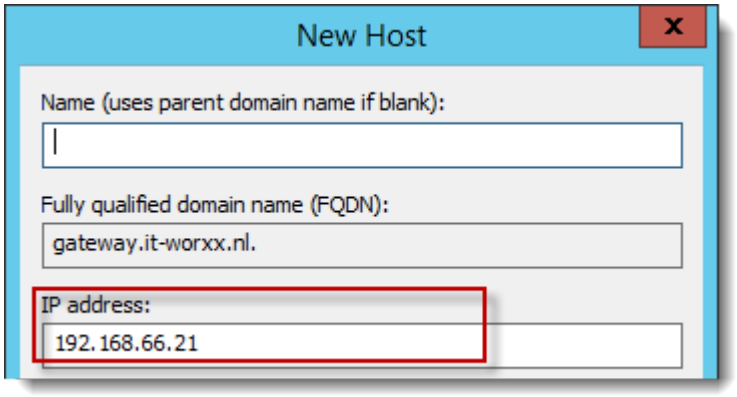
Finish the rest of the wizard accepting the defaults.

Browse to the newly created zone.



Right click the newly created zone and click New Host (A or AAAA)…

New Host



Leave the Name field blank, but enter the member server's (holding the RD Connection Broker role) IPv4 address.

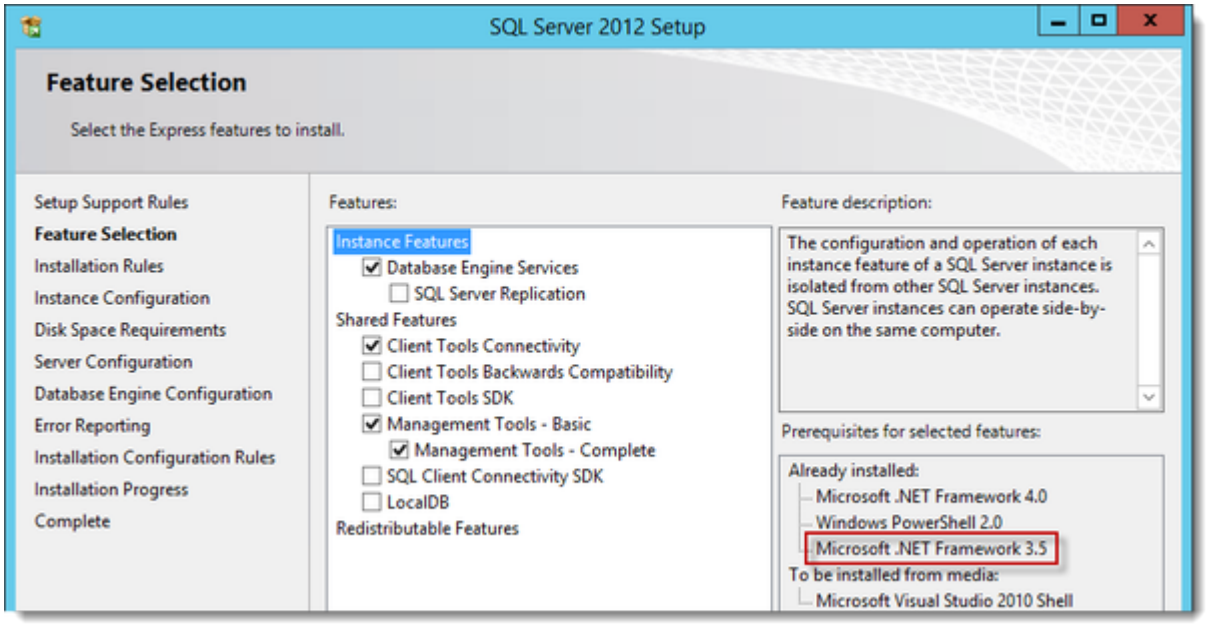
Click Add Host.

Create a new Global Security Group called "RDS Connection Brokers" and add the computer account for the member server to it as a group member.

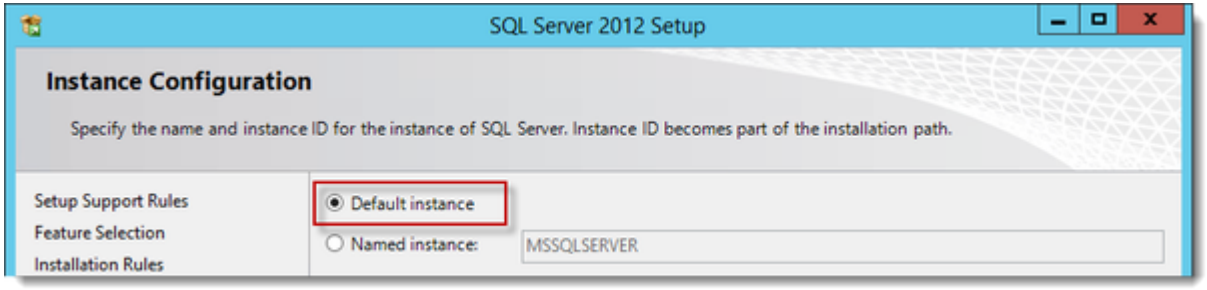
We need this group to be able to convert the RD Connection Broker to a highly available RD Connection Broker. You'll see why we need to do this in a few steps.

Reboot the member server to let it know it's a member of the RDS Connection Brokers security group.

Install SQL Express on the Domain Controller (or use an existing SQL Server if you already have one). Here's a list of needed features:

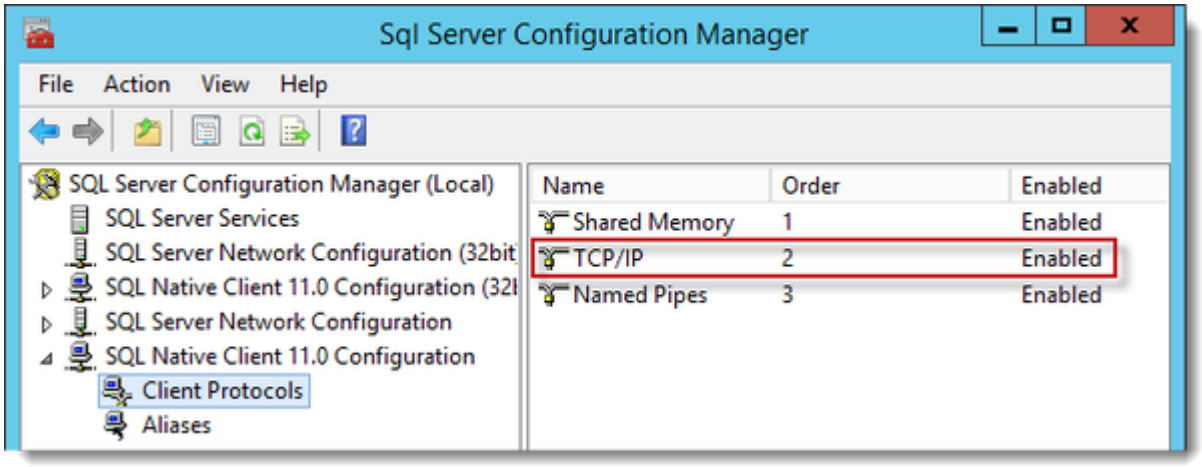


Now you see why I pre-configured the servers with the .NET Framework 3.5 feature before starting anything.



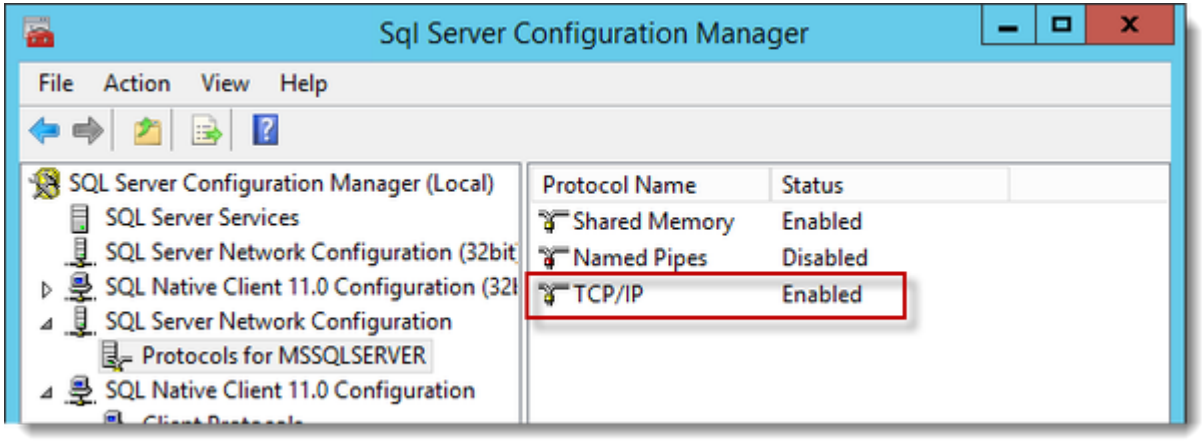
Use the Default Instance (so click Default, and do not leave the wizard's selection on Named instance: SQLEXPRESS).

When the installation is done open SQL Configuration manager and browse to Client Protocols under SQL Native Client 11.0 Configuration.



Check if TCP/IP is enabled under Client Protocols. SQL Express install enables this by default, but check it just to be sure, especially if you use an existing SQL Server.

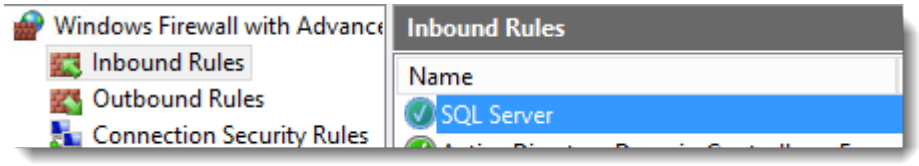
Browse to Protocols for MSSQLSERVER under SQL Server Network Configuration.



Enable TCP/IP. If this is a new SQL installation, this will be disabled by default.

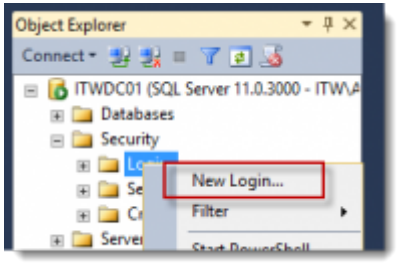
Restart the SQL Server service if you changed this setting.

On the SQL Server, make sure port 1433 is not being blocked by Windows Firewall.



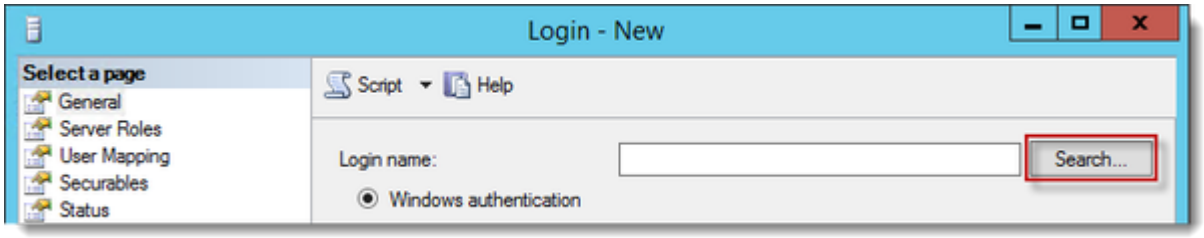
I added the SQL Server executable to the exception list to allow all inbound traffic.

Open SQL Server Management Studio and browse to Logins under Security.



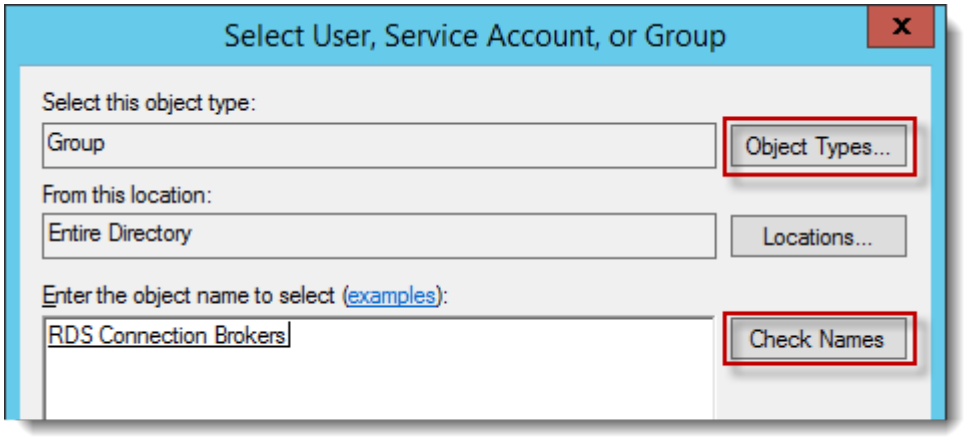
Right click Logins and click New Login…

Login – New



Click Search…

Select User, Service Account, or Group

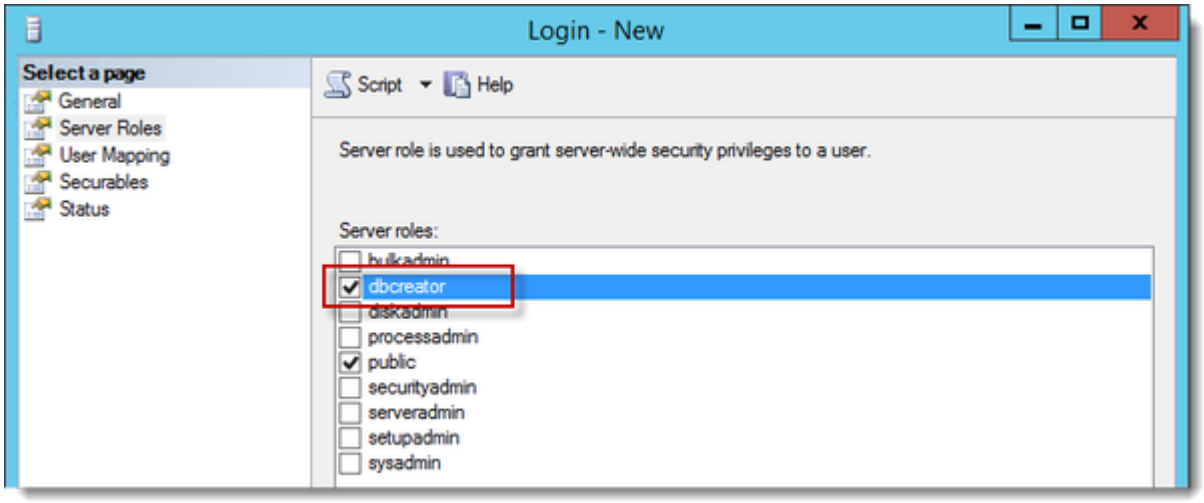


Click Object Types… and select Group.

Type the RDS Connection Brokers security group name and click Check Names.

Click OK.

Login – New



Click Server Roles and select dbcreator.

Click OK.

We have just effectively granted the RDS Connection Broker server the right to create databases.

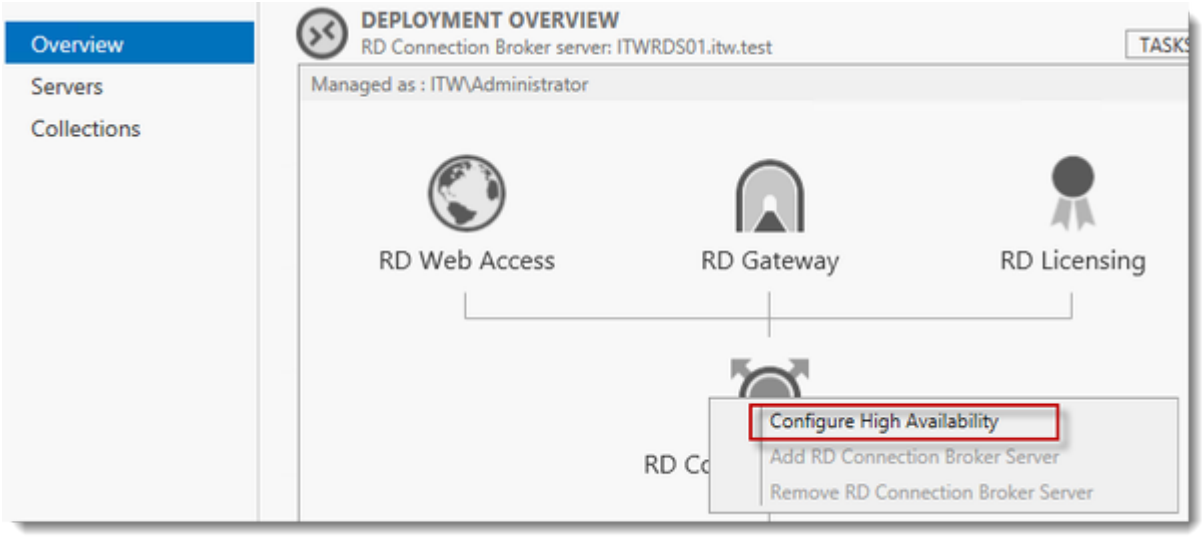
We need this because the RDS Connection Broker service will try to migrate from WID (Windows Internal Database to a (high available) SQL Server instance when we convert the Broker to a high available broker.

Install the SQL Native Client on the member server (Client Components only).

Everything we need is now in place to convert the RD Connection Broker, so let's do just that.

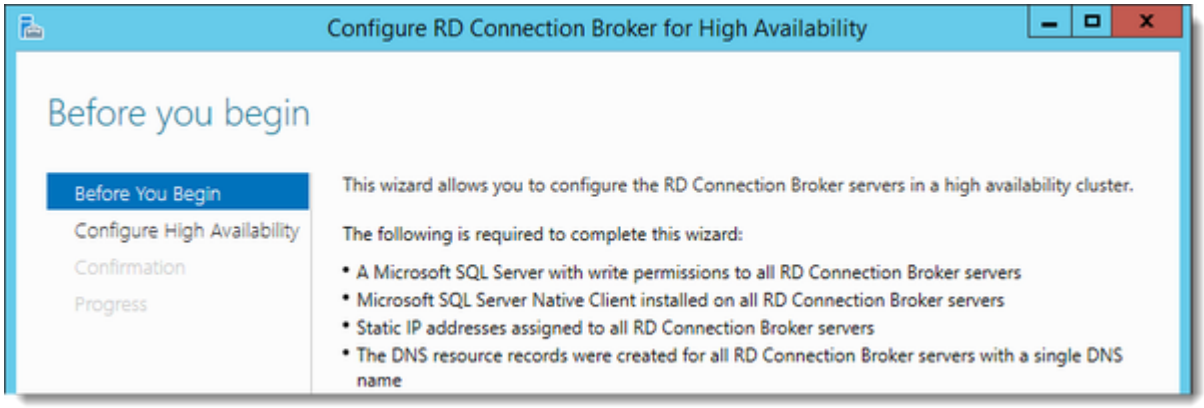
Convert the RD Connection Broker

In Server Manager click Remote Desktop Services and scroll down to the overview.



Right click RD Connection Broker and click Configure High Availability.

Before you begin



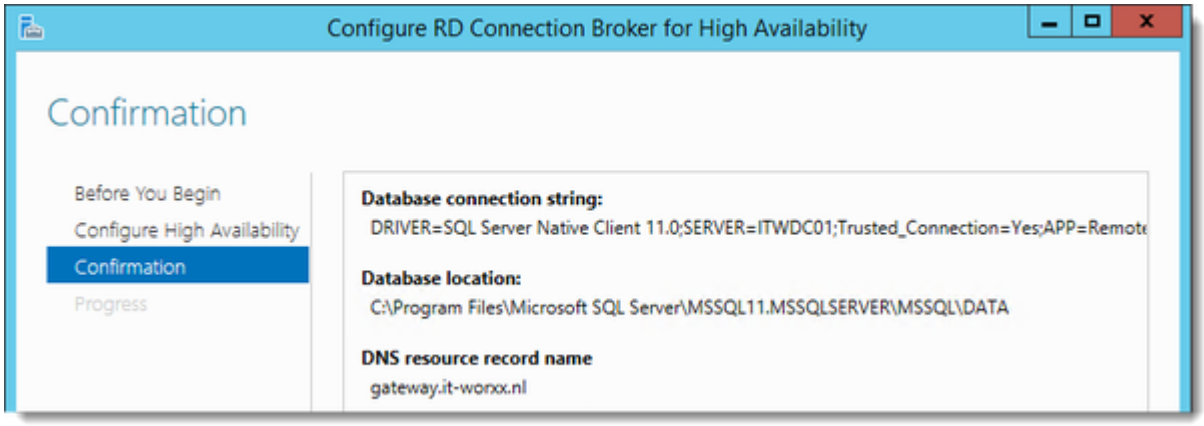
So we're actually building a single node cluster here ;)

Look at the pre-requisites.

If you have more than one RD Connection Broker they need to be configured using DNS Round Robin. More on that in a later post.

Click Next.

Configure RD Connection Broker for High Availability



Database connection string:

DRIVER=SQL Server Native Client 11.0;SERVER=ITWDC01;Trusted_Connection=Yes;APP=Remote Desktop Services Connection Broker;DATABASE=ITWRDCB

Folder to store database files:

C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA

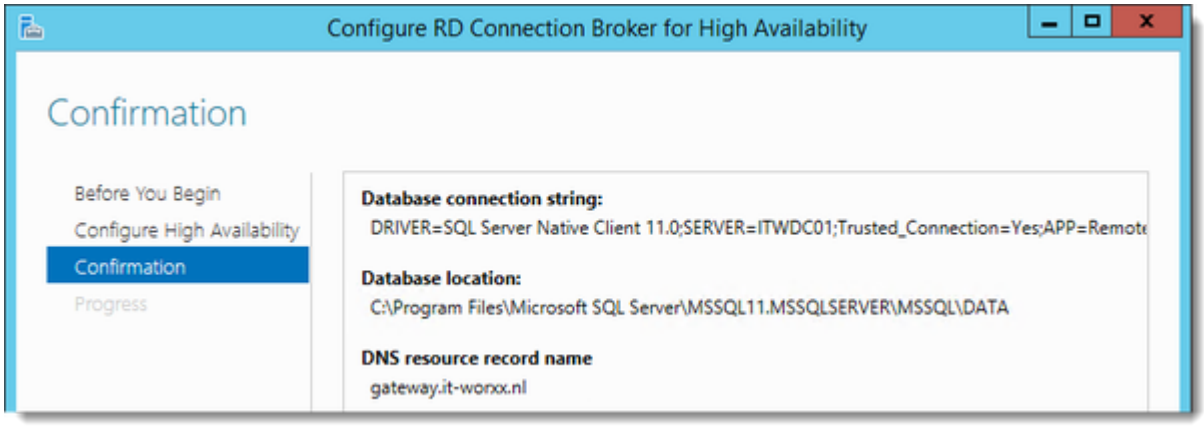
I used the instance default folder.

DNS round robin name:

The DNS Zone name we configured in DNS earlier.

Click Next.

Confirmation



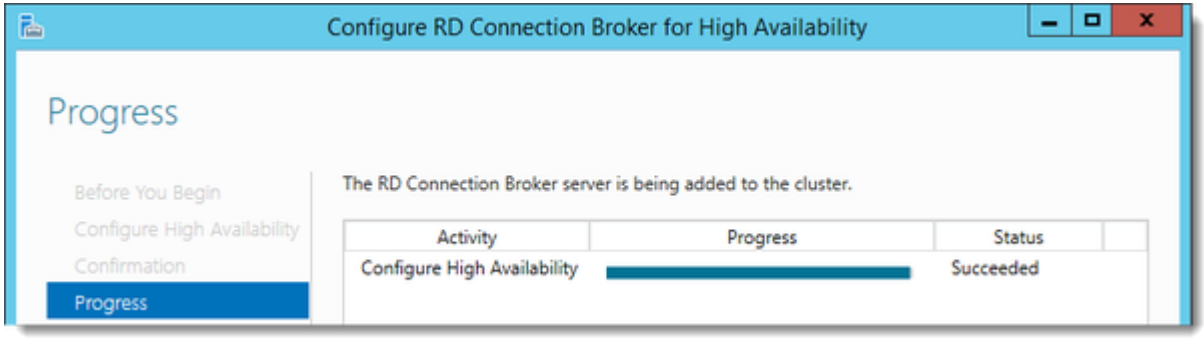
If you get an error before this page:

– Check if TCP/IP is enabled in client protocols and for your instance

– Check if you can reach port 1433 on the SQL Server from the member server

Click Configure.

Progress

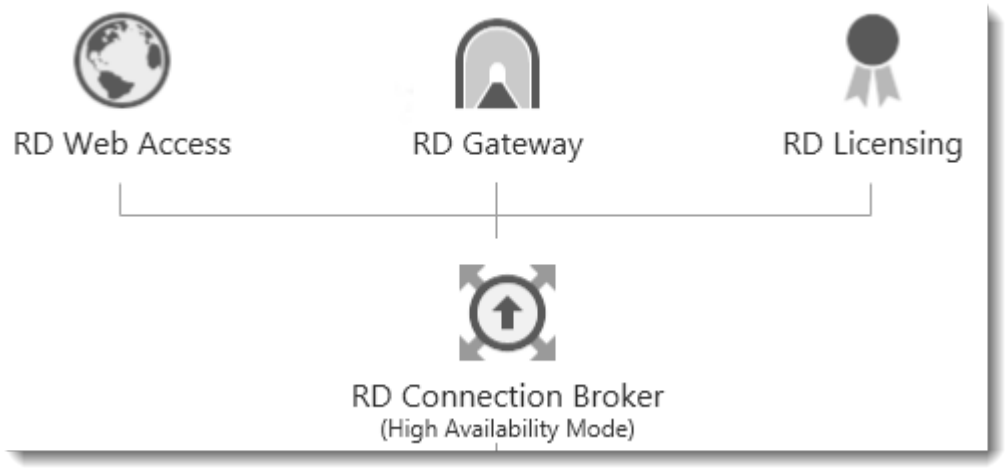


If you get an error on this page:

– Check SQL permissions for the security group

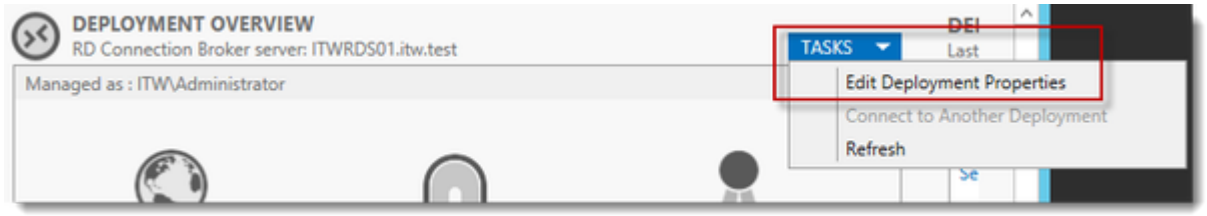
– Check if the database path you entered is correct

Click Close.



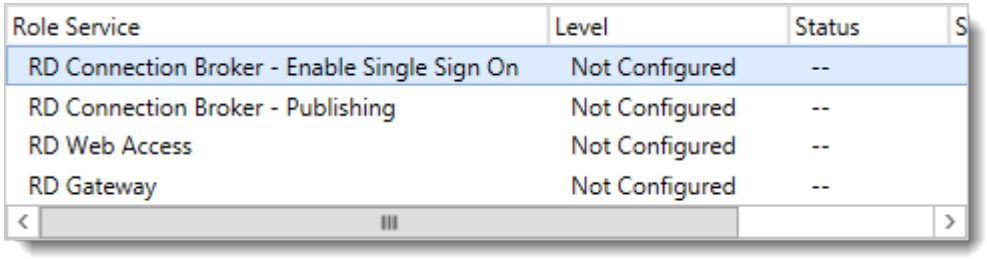
The RD Connection Broker is now in High Availability Mode and we are finally ready to complete the configuration.

Completing the Remote Desktop Services configuration

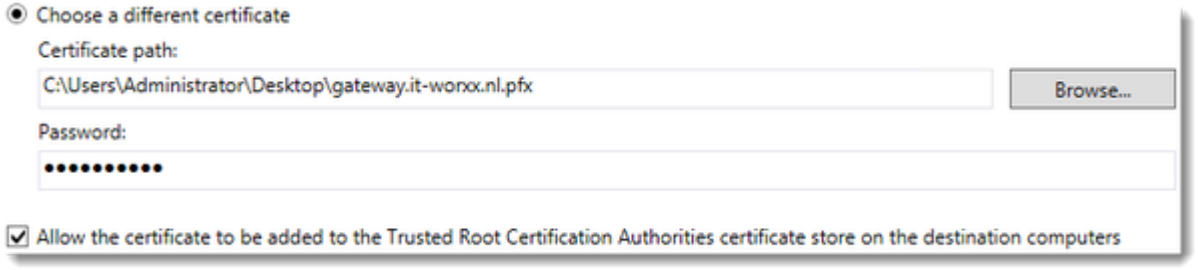


In Server Manager, Remote Desktop Services, Overview, click Tasks and click Edit Deployment Properties, then click Certificates.

Configure the deployment

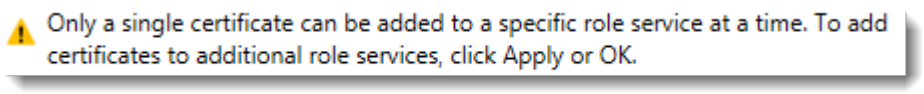


Click RD Connection Broker – Enable Single Sign On and click Select Existing certificate.



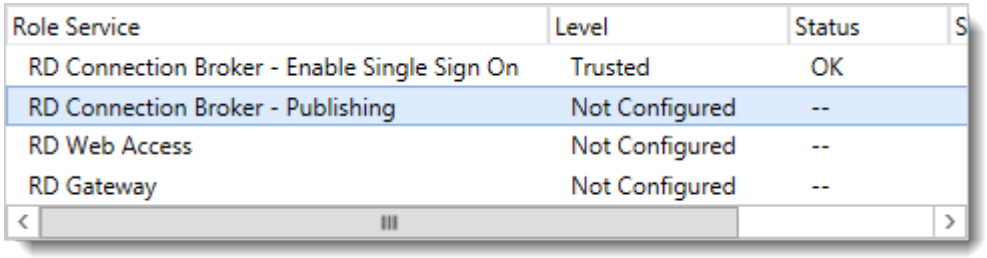
Browse to the .pfx file, enter its password, and check Allow the certificate..

Click OK.



So click Apply. This takes a little while, be patient.

Configure the deployment



Click RD Connection Broker – Publishing and click Select Existing certificate.

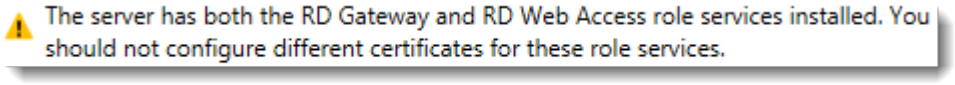
Browse to the .pfx file, enter its password, and check Allow the certificate..

Click OK.

Click Apply. This again takes a little while, be a little more patient.

Configure the deployment

Click RD Web Access and click Select Existing certificate.



Note: Did you notice the warning when you select RD Web Access? Browse to the .pfx file, enter its password, and check Allow the certificate..

Click OK.

Click Apply again. This takes another little while longer, be a slightly more patient.

Configure the deployment

Last one. Click RD Gateway and click Select Existing certificate.

Browse to the .pfx file, enter its password, and check Allow the certificate..

Click OK.

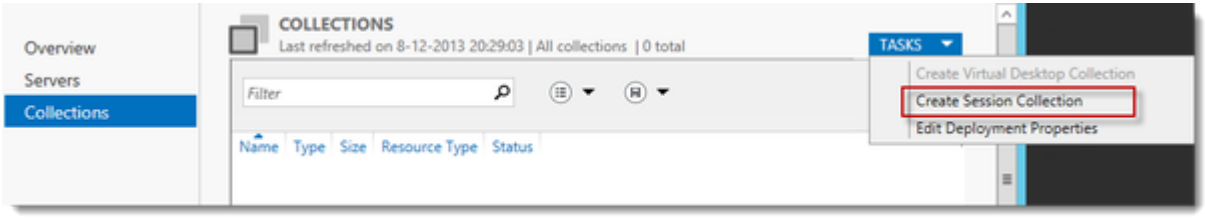
Click OK to finish the certificate configuration.

Configured all servers, configured certificates..

One thing left to do: Tell our RDS environment exactly what to publish.

In fact you can use this setup to either provide full desktop sessions on the Session Host, or you can choose to publish only applications on the Session Host.

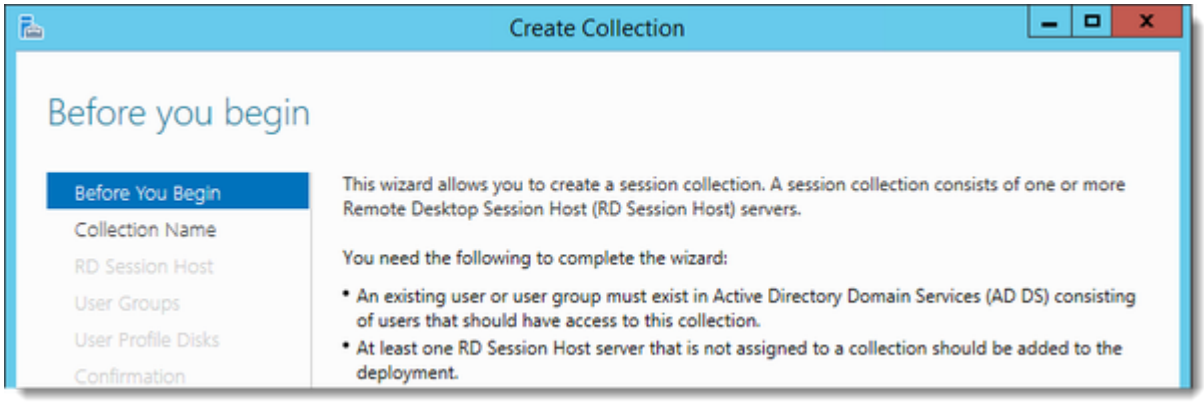
Let’s publish full desktop sessions.



Publish a full Remote Desktop environment

In Server Manager, Remote Desktop Services, Session Collections, click Tasks and click Create Session Collection.

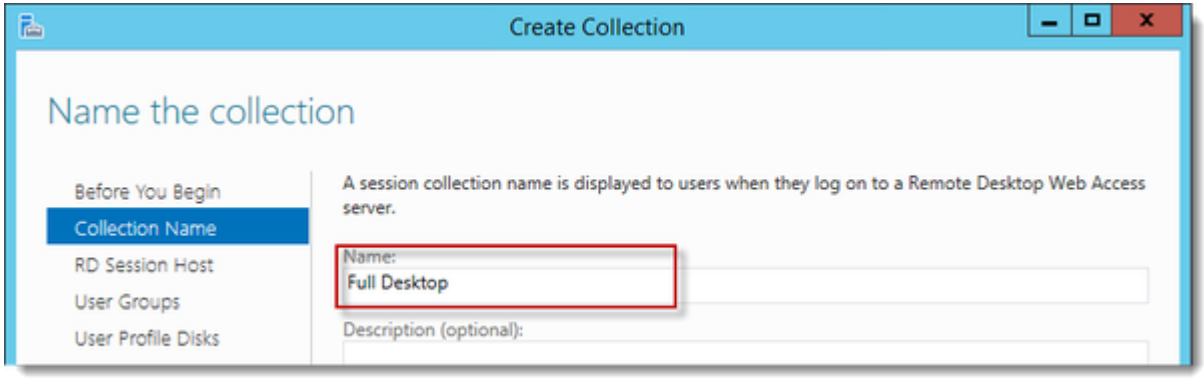
Before you begin



Review the requirements. This won't be an issue in this setup, but you could restrict access to this collection by selecting a select group of people.

Click Next.

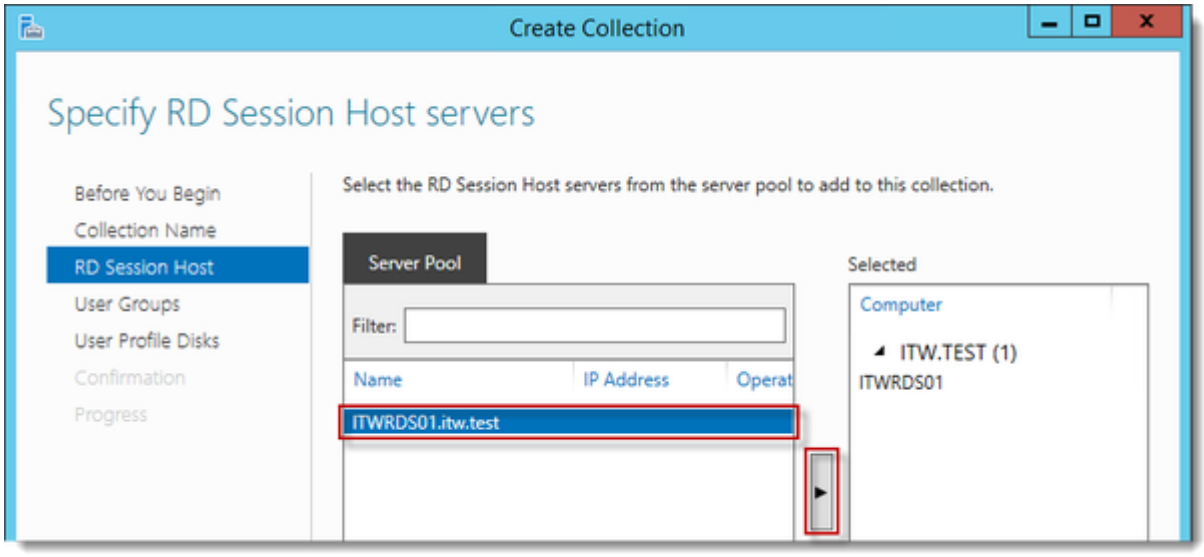
Name the collection



Enter a descriptive name. This name will be displayed under its icon in the Web Access interface.

Click Next.

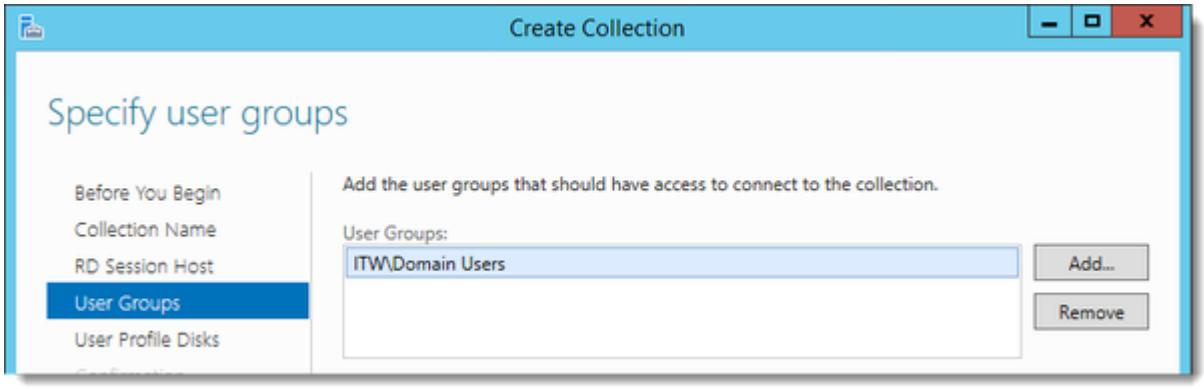
Specify RD Session Host servers



Click the member server and click the Add button.

Click Next.

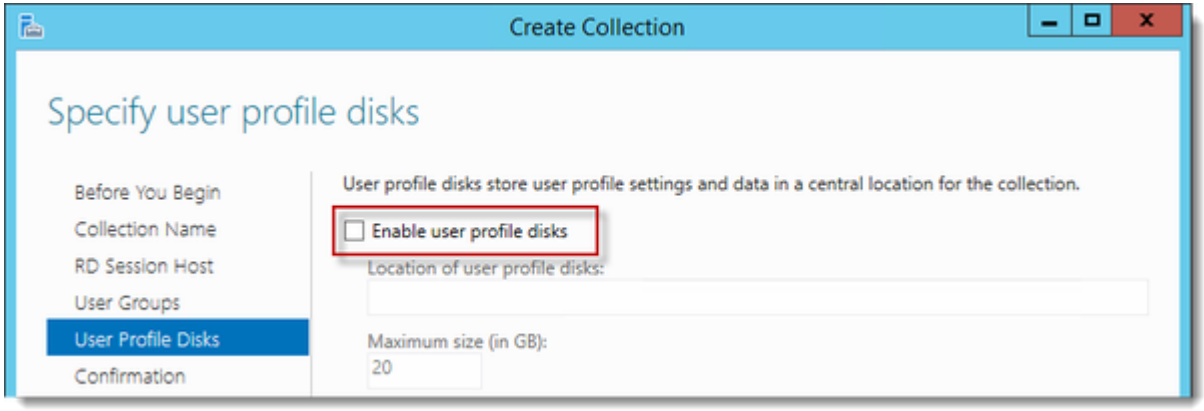
Specify user groups



You can limit access here. Add one or more groups to restrict access to these groups only. In this setup Domain Users will do fine.

Click Next.

Specify user profile disks

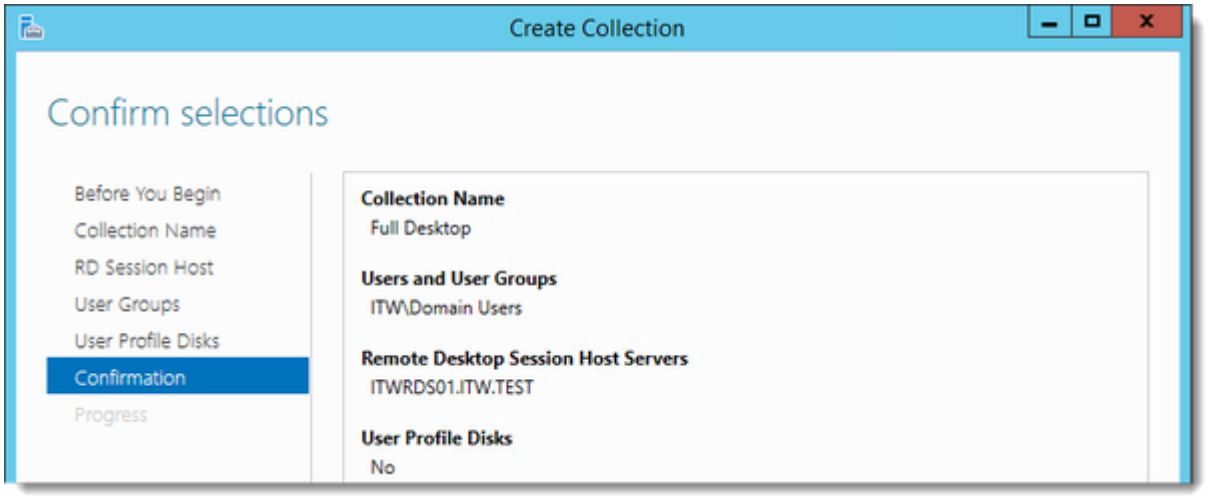


User profile disks are not in focus in this guide. Since I have no file shares configured in this setup, uncheck Enable user profile disks for now.

Does and Don’ts will be covered in a future post.

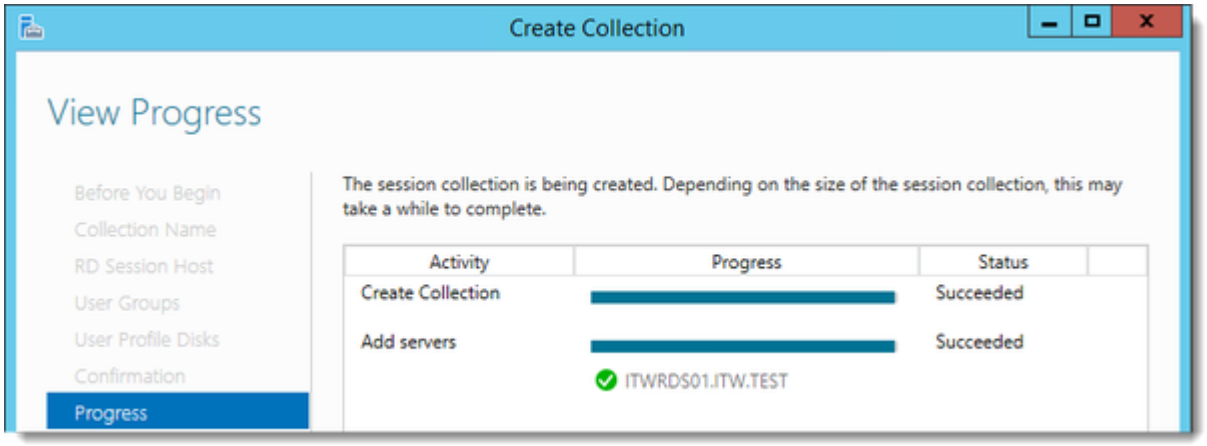
Click Next.

Confirm selections



Review the information and click Create.

View Progress



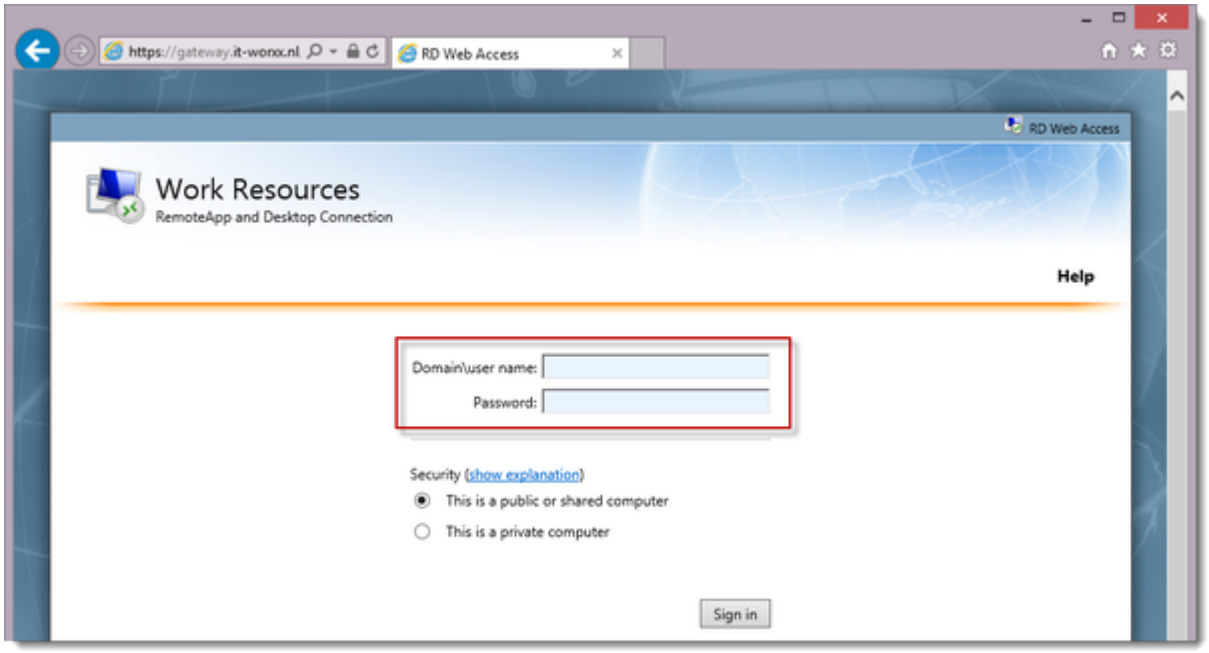
Wait until the collection is created and the server is added to the collection.

Click Close.

Time to test the setup!

Testing the Remote Desktop Services

On a machine that has access to your test setup (you may have to add the external FQDN to your hosts file if you didn't publish it to the internet) open https://gateway.it-worxx.nl/rdweb.

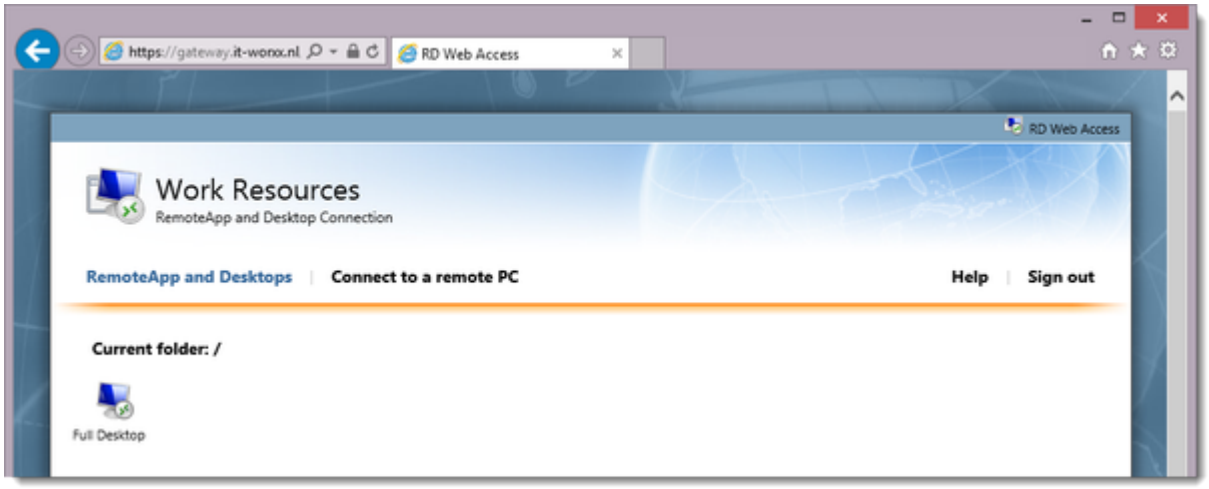


Hey! At least the RD Web Access application works :)

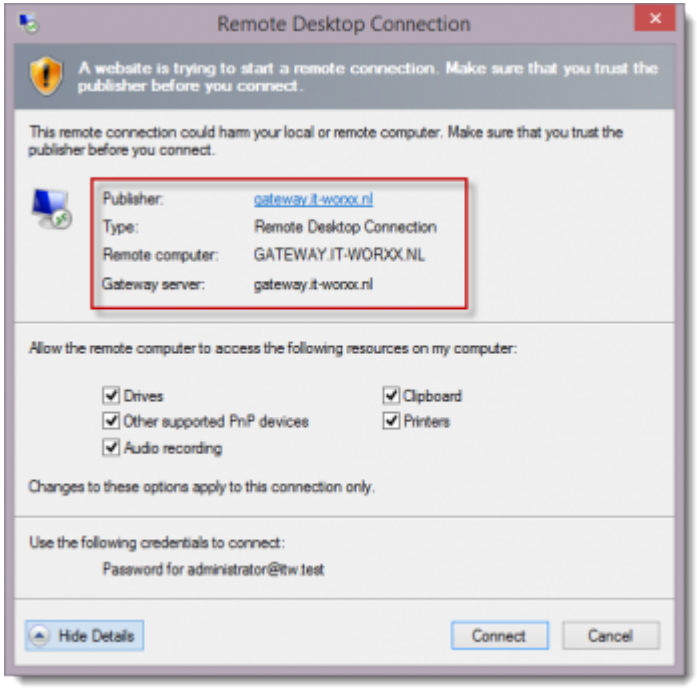
Enter a valid username and password (ITW\username or username@itw.test).

Create a user for this, or simply use the domain admin account.

Click Sign in.



After logging in you're presented with the full desktop session collection we created.



After clicking the Full Desktop icon you get the warning that devices are going to be redirected.

And when you click Connect, you actually connect :)

Enjoy.

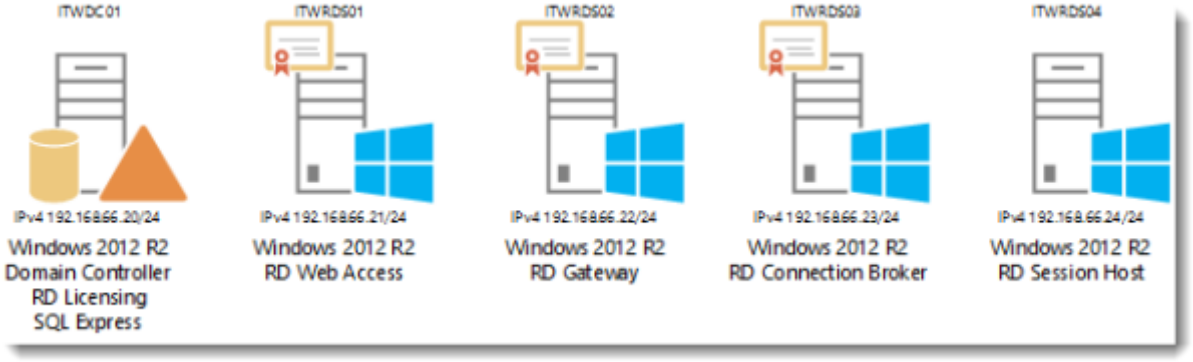
Step by Step Windows 2012 R2 Remote Desktop Services – Part 2

A step by step guide to build a Windows 2012 R2 Remote Desktop Services deployment.

Part 2 – Deploying an advanced setup.

In part one I detailed how to do a single server installation. In case you missed it, or want to check it out, look at this post:
https://msfreaks.wordpress.com/2013/12/09/windows-2012-r2-remote-desktop-services-part-1/

In this step by step guide we'll be building a more complex setup:



As you can see we'll deploy 3 certificates in this setup. The names I will use for this will be "webaccess.it-worxx.nl", "gateway.it-worxx.nl" and "broker.it-worxx.nl" for obvious reasons. You may consider using a wildcard certificate.

Software used in this guide: Windows Server 2012 R2 ISO (evaluation can be downloaded here: http://technet.microsoft.com/en-us/evalcenter/dn205286.aspx)

SQL Server 2012 SP1 Express x64 With tools (free version can be downloaded here:http://www.microsoft.com/en-us/download/details.aspx?id=35579. After clicking the download button select SQLEXPRWT_x64_ENU.exe)

SQL Server 2012 SP1 Native Client (free version can be downloaded here:http://www.microsoft.com/en-us/download/details.aspx?id=35580. After clicking the download button select ENU\x64\sqlncli.msi)

And three certificates. I got mine for free from https://startssl.com. The certificate need to contain the FQDNs you will use for publishing the RD Web Access (webaccess.it-worxx.nl) and RD Gateway (gateway.it-worxx.nl) roles. You'll also need one for the RD Broker role, even though we won't publish this server to the internet. The files need to be in .pfx format and you need to have the private key in them.

As in the previous guide, this guide will not focus on building a domain using a single domain controller and adding the other servers as member servers to this domain.

And again some basic knowledge is assumed in this guide.

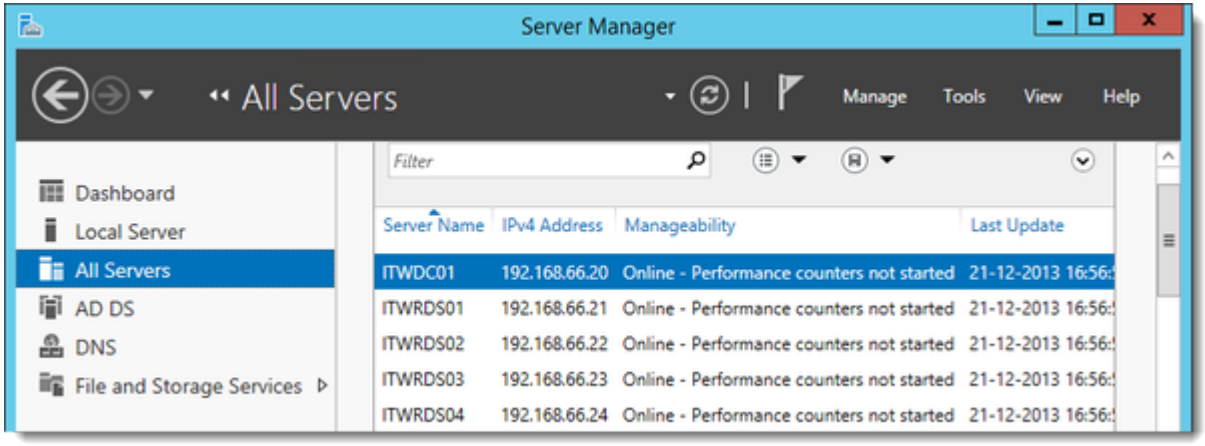
I will be using Hyper-V 3.0 on my Windows 8.1 laptop and I have prepared 5 servers. The servers will be similar to the 2 I used in the previous guide. All servers have the .NET Framework 3.5 added as a feature.

All servers have 1vCPU, 512MB memory, and a dynamic 60GB Harddisk) I configured ITWDC01 as a Domain Controller in a new forest: itw.test.

I added the rest of the servers as member servers to the itw.test domain and configured them to use ITWDC01 as their primary DNS server.

Installing the Remote Desktop Services Roles

Log on to the Domain Controller, and in Server Manager right-click the All Servers node and add all other servers using the Add Servers command (or select the All Servers node, click Manage and click Add Servers).

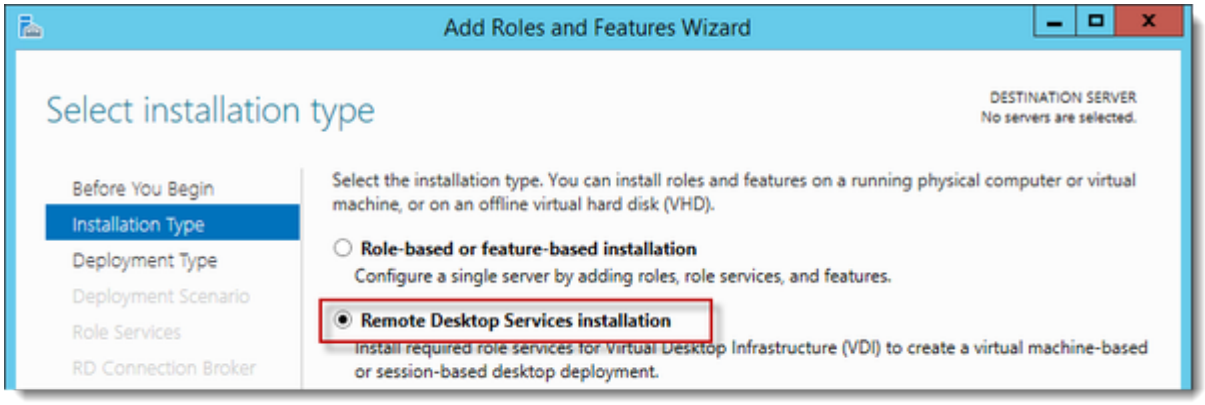


Now that all servers needed in this deployment scenario are present, click Manage, and click Add Roles & Features.

Before you begin

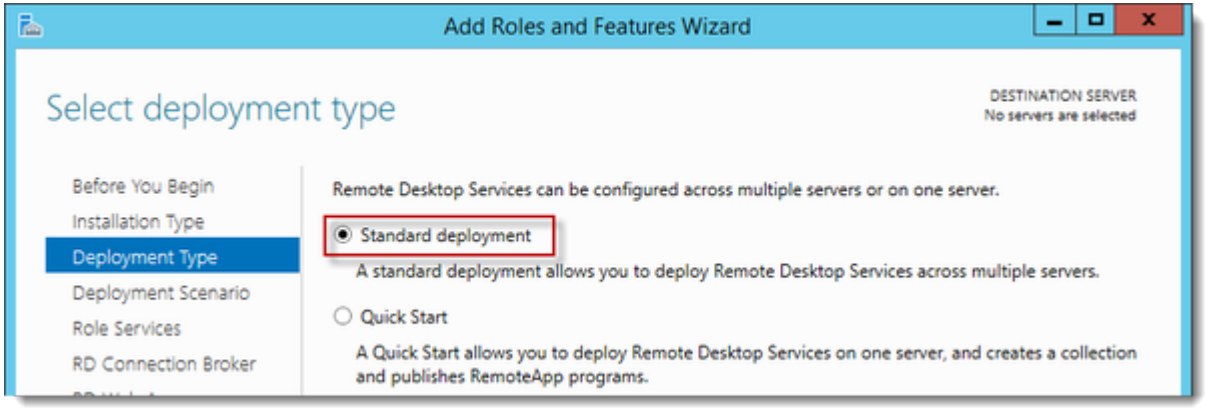
Click Next.

Select Installation Type



Select Remote Desktop Services installation. Click Next.

Select Deployment Type



Select Standard deployment.

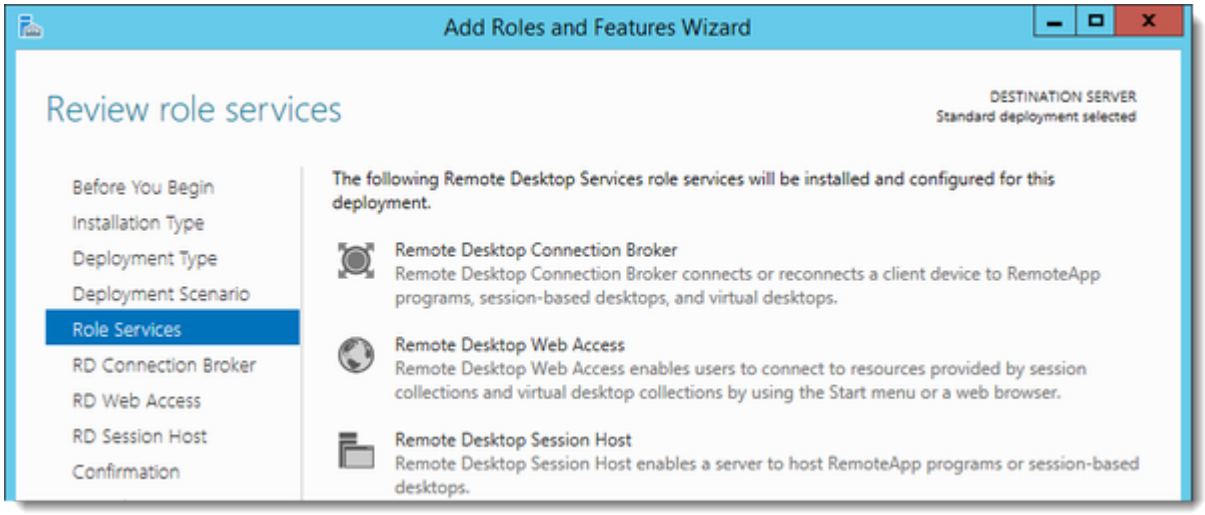
Click Next.

Select Deployment Scenario

Select Session-based desktop deployment. The other option will be a different post in this series.

Click Next.

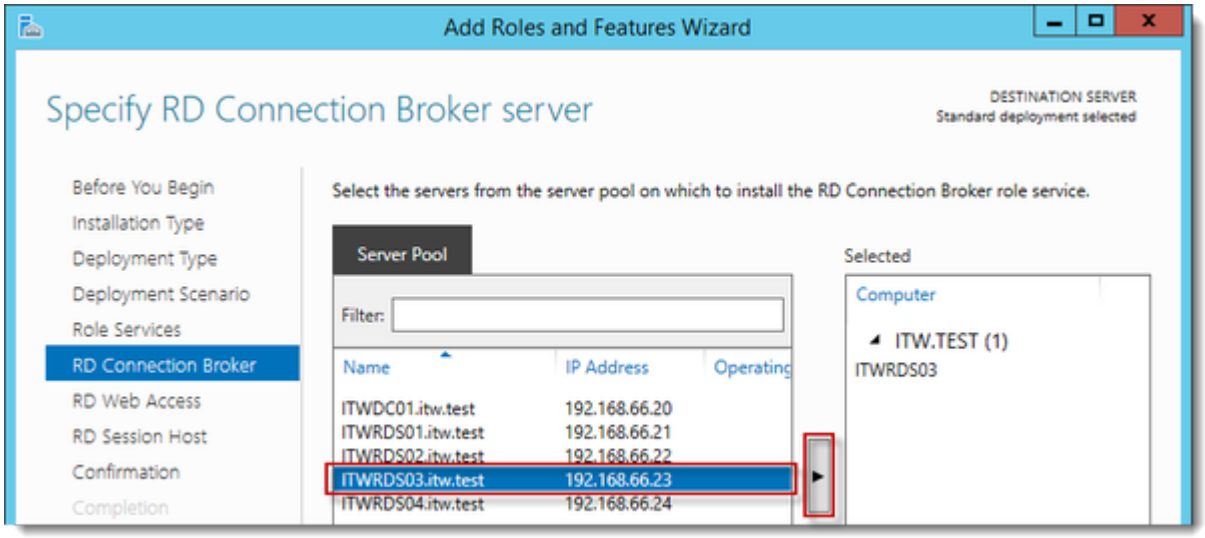
Review Role Services



Review the services that will be installed.

Click Next.

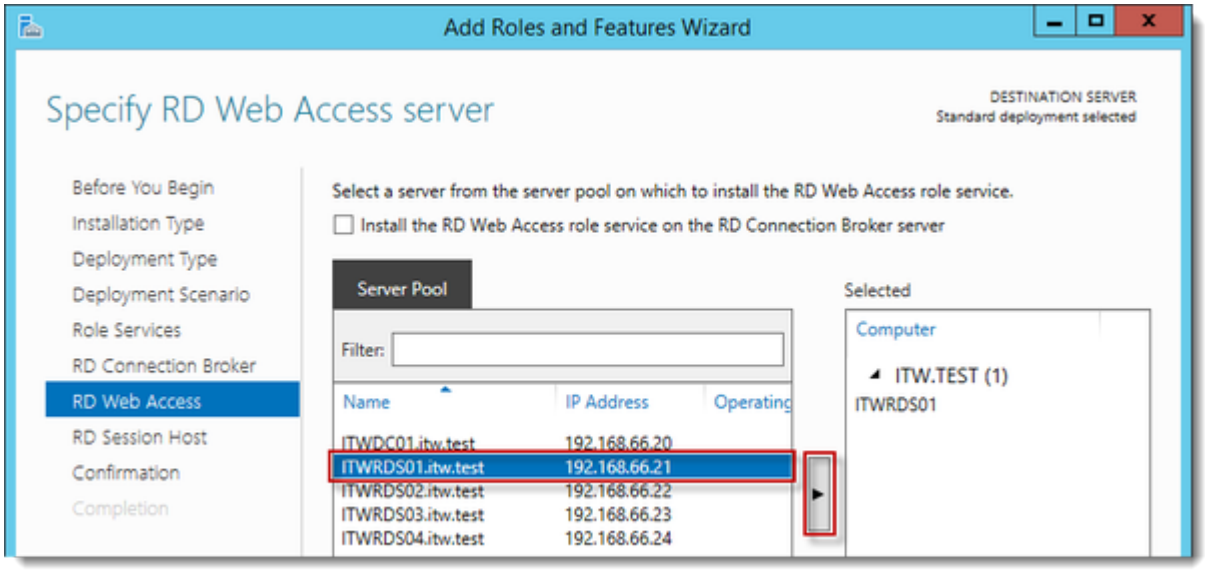
Specify RD Connection Broker server



Click the preferred server and click the Add button.

Click Next.

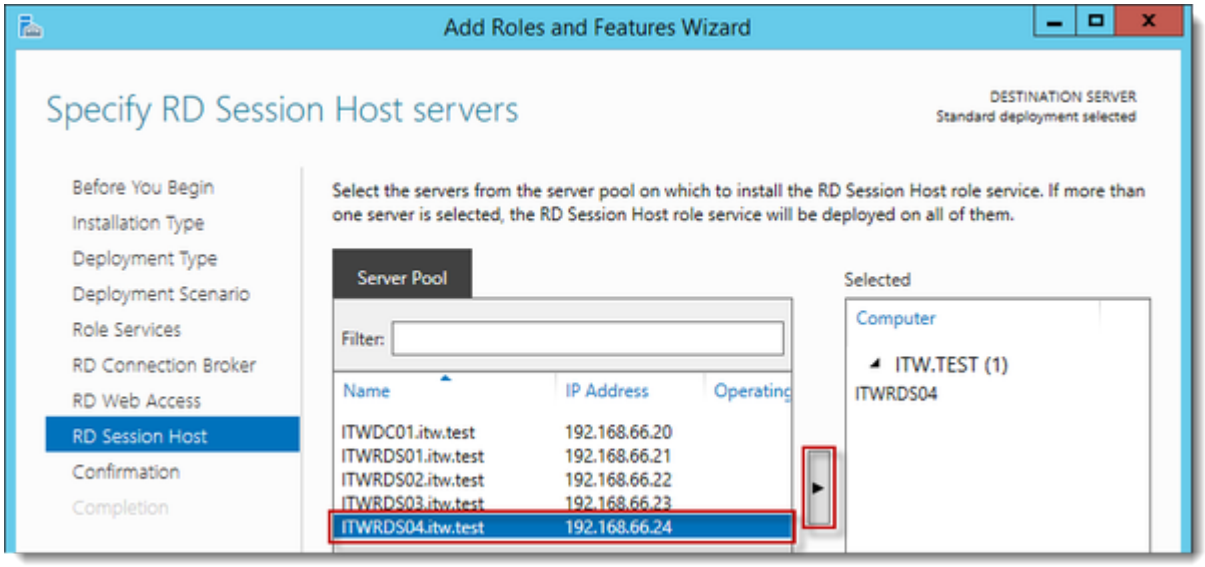
Specify RD Web Access server



Click the preferred server and click the Add button.

Click Next.

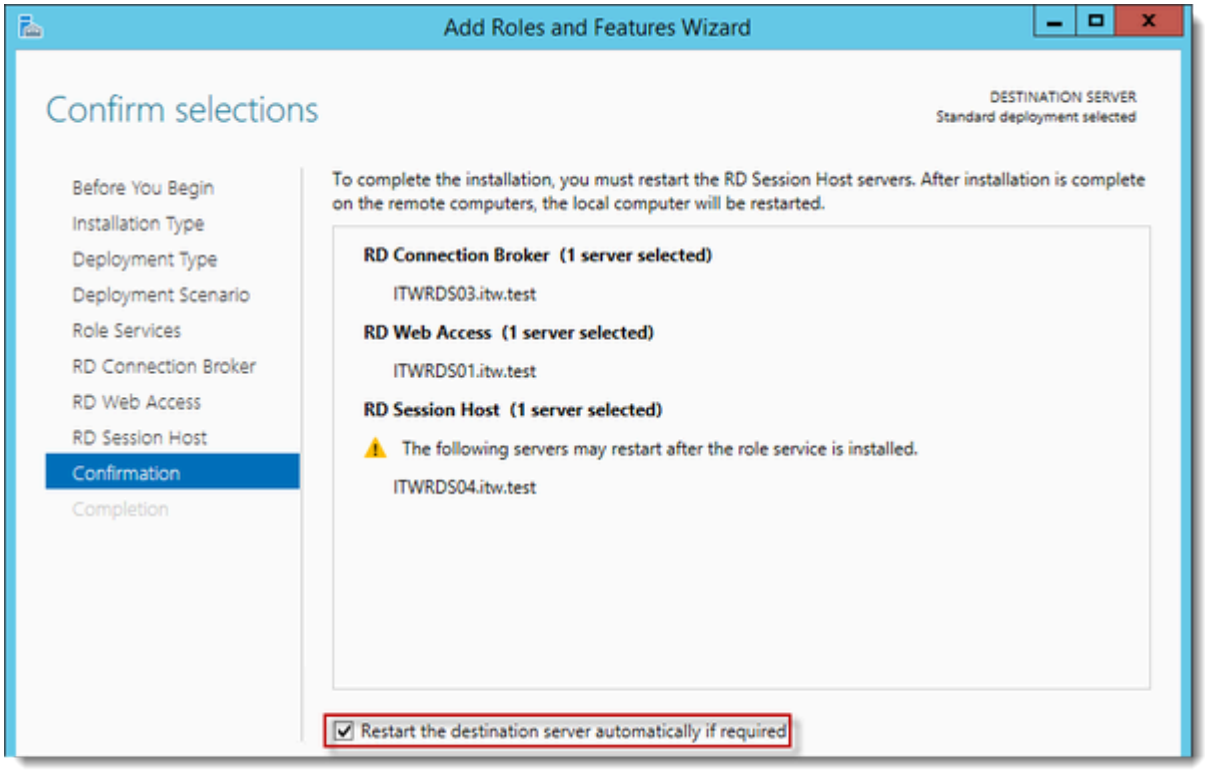
Specify RD Session Host server



Click the preferred server and click the Add button.

Click Next.

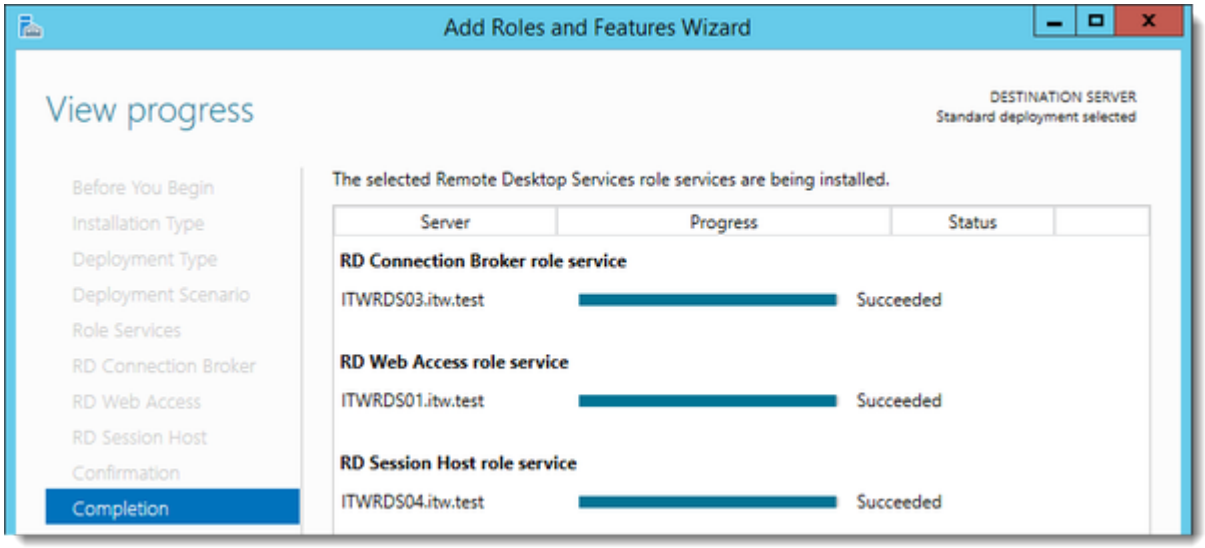
Confirm selections



Check Restart the destination server automatically if required.

Click Deploy.

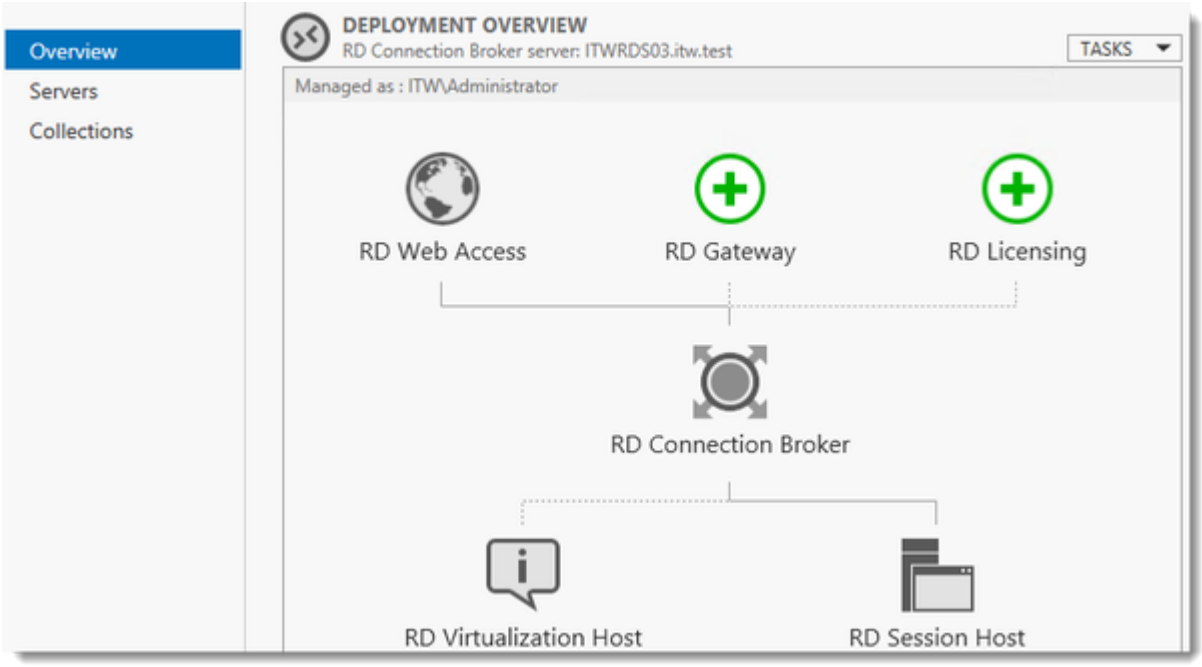
View progress



Wait until all role services are deployed and the RD Session Host server has restarted.

Click Close.

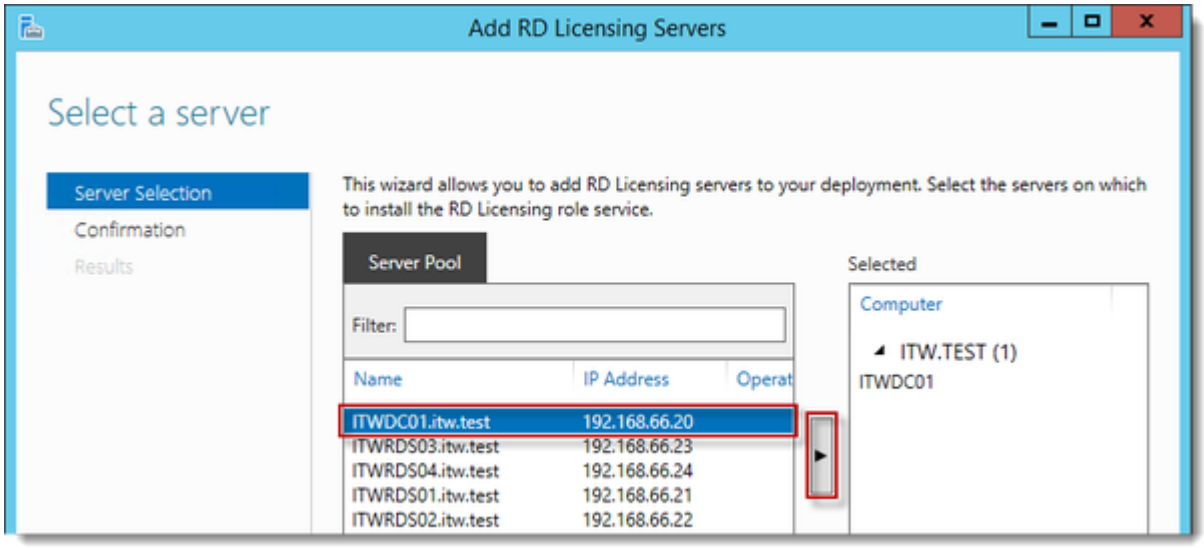
In Server Manager click Remote Desktop Services and scroll down to the overview.



As you can see the deployment is missing a RD Gateway server and a RD Licensing server.

Click the Add RD Licensing server button.

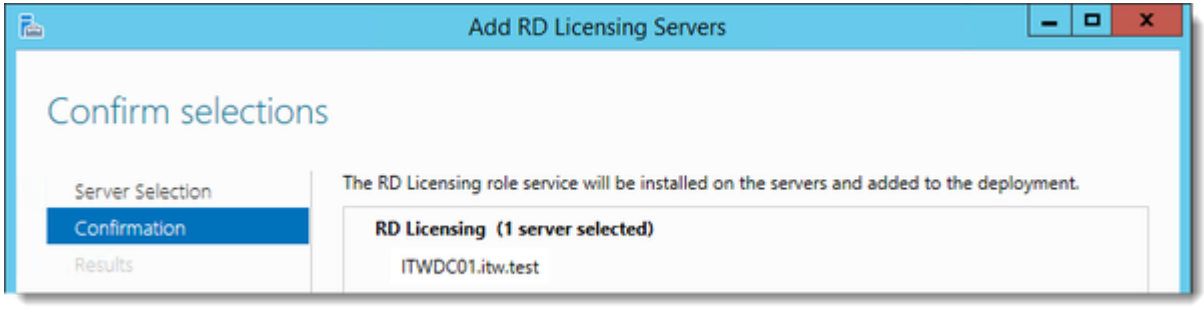
Select a server



Click the domain controller and click the Add button.

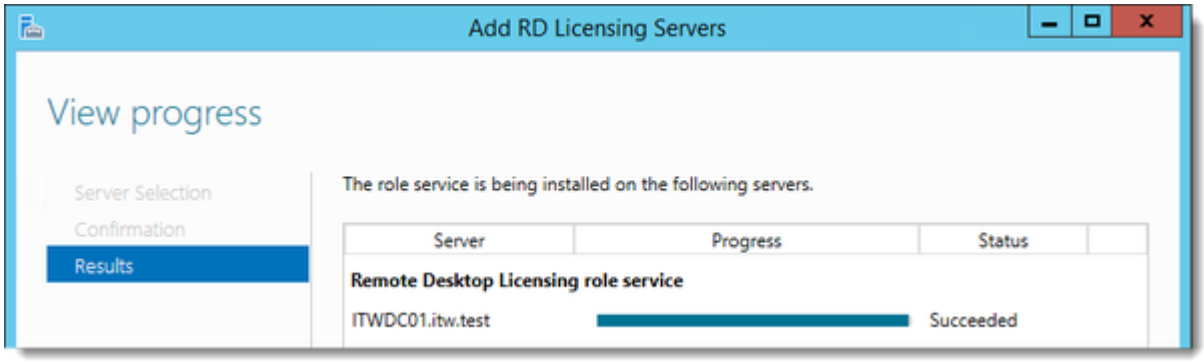
Click Next.

Confirm selections



Click Add.

View progress

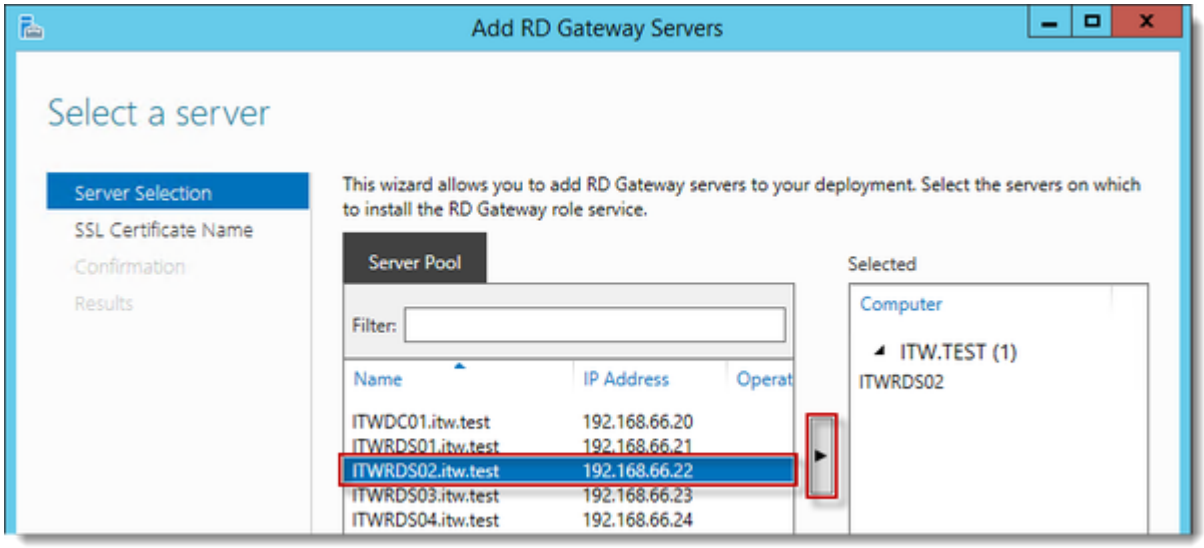


Wait until the role service is deployed. No restart is needed.

Click Close.

Click the Add RD Gateway server button.

Select a server



Click the correct server and click the Add button.

Click Next.

Name the self-signed SSL certificate

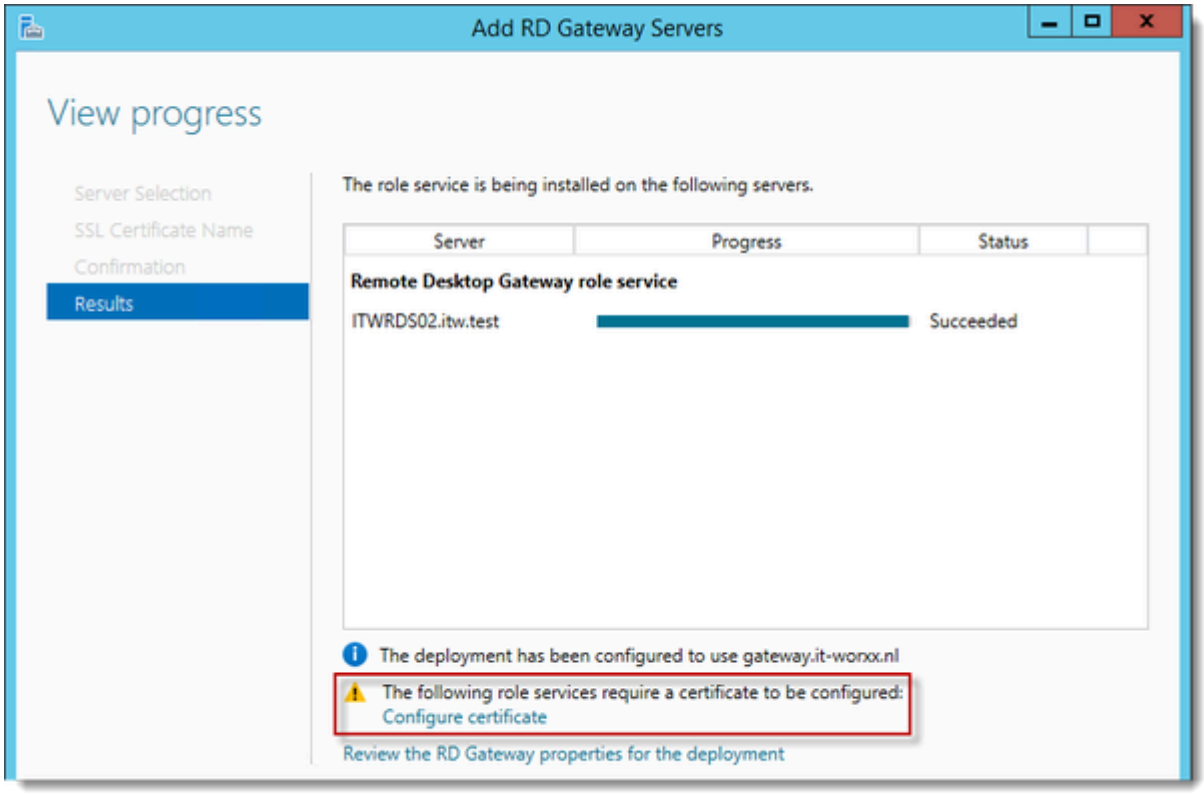
The wizard creates a self-signed certificate. We will deal with certificates in this deployment in a little bit. Enter the external Fully Qualified Domain Name for the Gateway URL. In my case, for lack of a better name, I used “gateway.it-worxx.nl.

Click Next.

Confirm selections

Click Add.

View progress



Wait until the role service is deployed. No restart is needed.

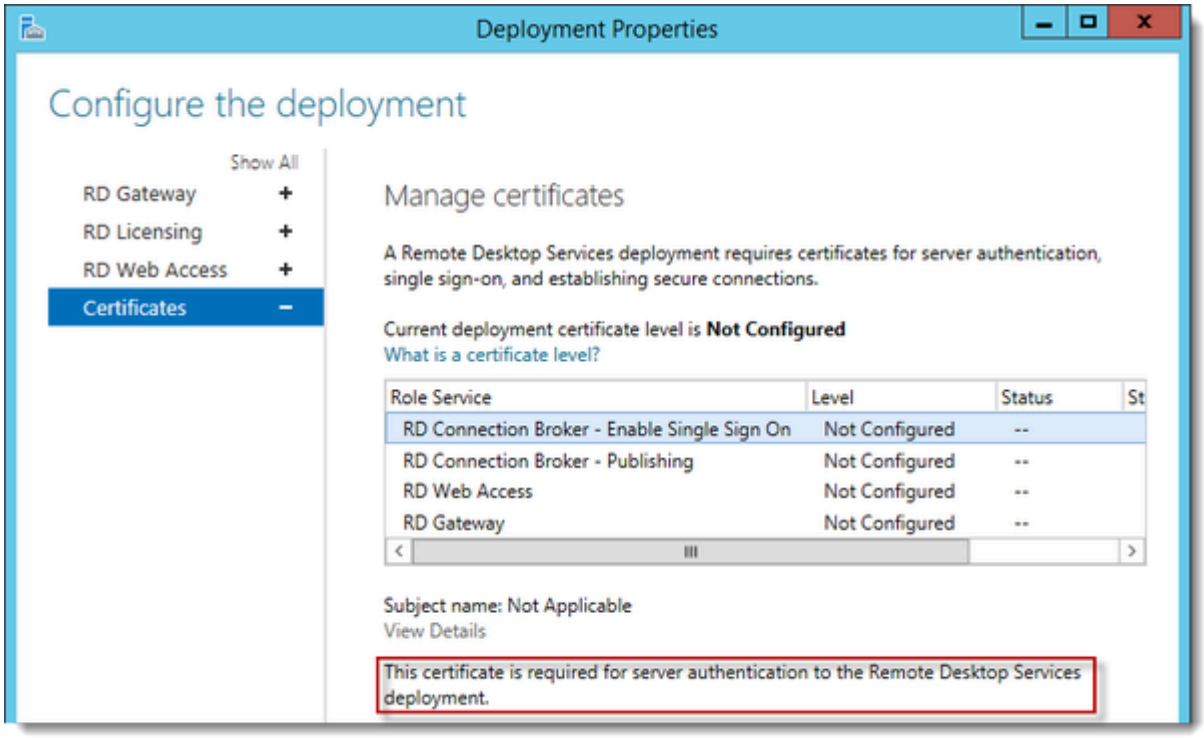
Notice that “gateway.it-worxx.nl” was configured for the deployment as a FQDN.

Also notice that certificate configuration is needed.

Notice the link in the bottom to “Review the RD Gateway properties for the deployment”.

Click Configure certificate.

Configure the deployment

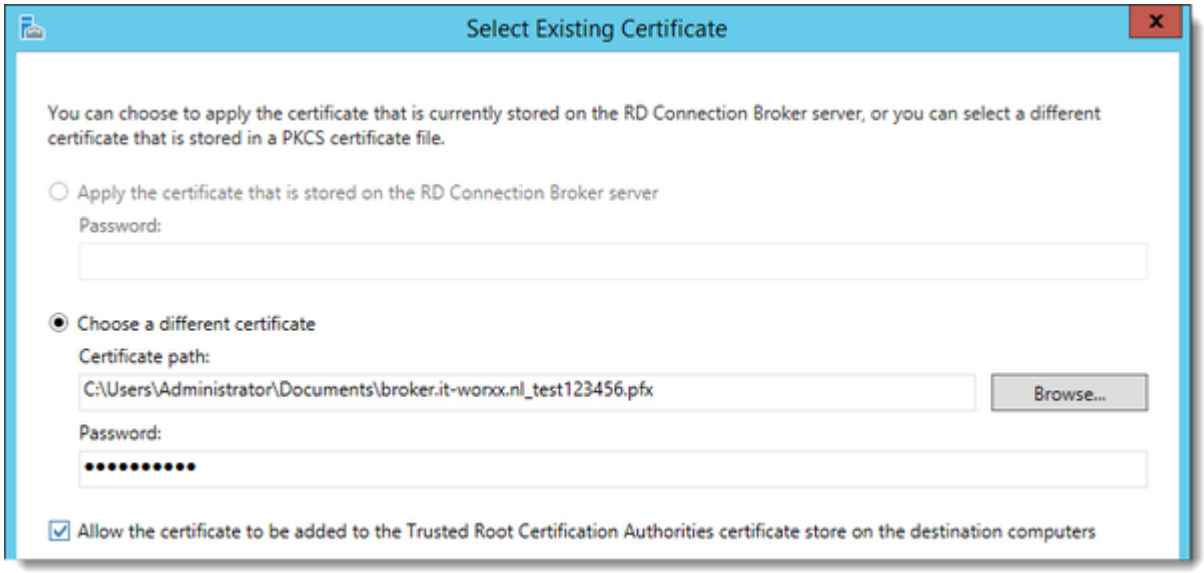


Click RD Connection Broker – Enable Single Sign On.

Notice the purpose of this certificate.

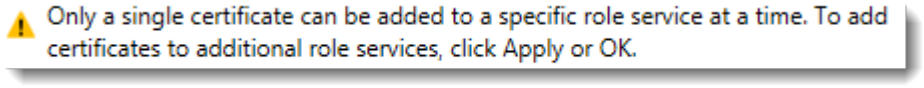
Click Select Existing Certificate.

Select Existing Certificate



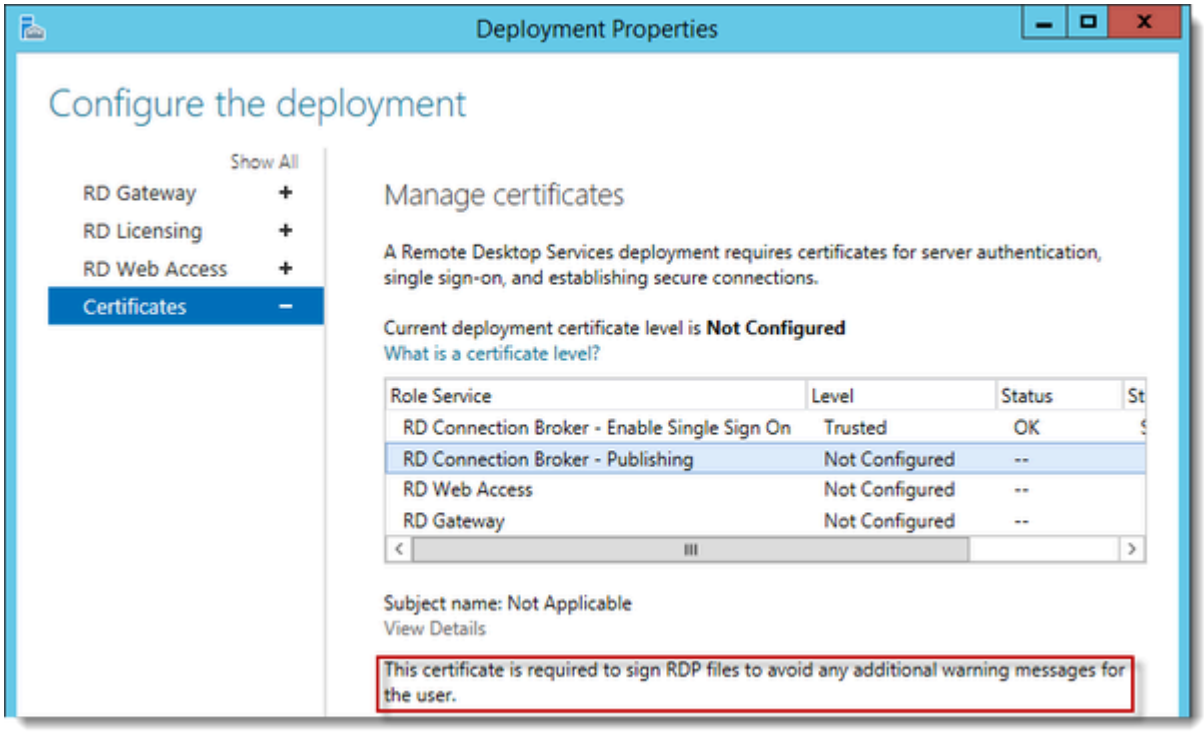
Click Browse to browse to the .pfx which you prepared for the RD Connection Broker server, enter the password for that .pfx and check “Allow the certificate to be added to the Trusted Root Certification Authorities certificate store on the destination computers”.

Click OK.



Click Apply to apply the certificate changes. Do not click OK because we need to configure the other certificate options as well and we can configure only one at a time.

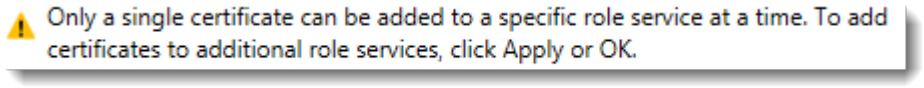
Configure the deployment



Select RD Connection Broker – Publishing.

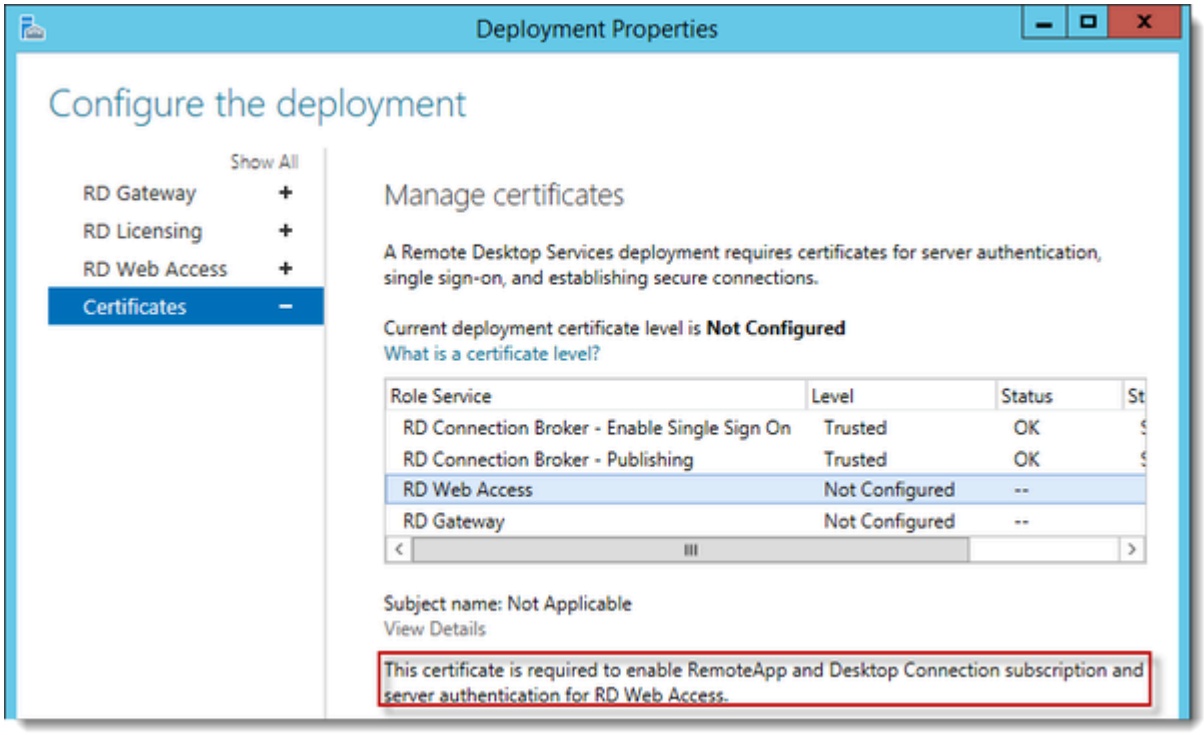
Notice the purpose of this certificate.

Click Select Existing Certificate and add the same certificate you added for RD Connection Broker – Enable Single Sign On.



Click Apply to apply the certificate changes. Do not click OK because we need to configure the other certificate options as well and we can configure only one at a time.

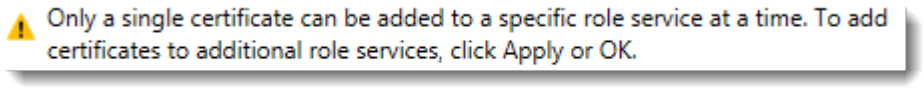
Configure the deployment



Select RD Web Access.

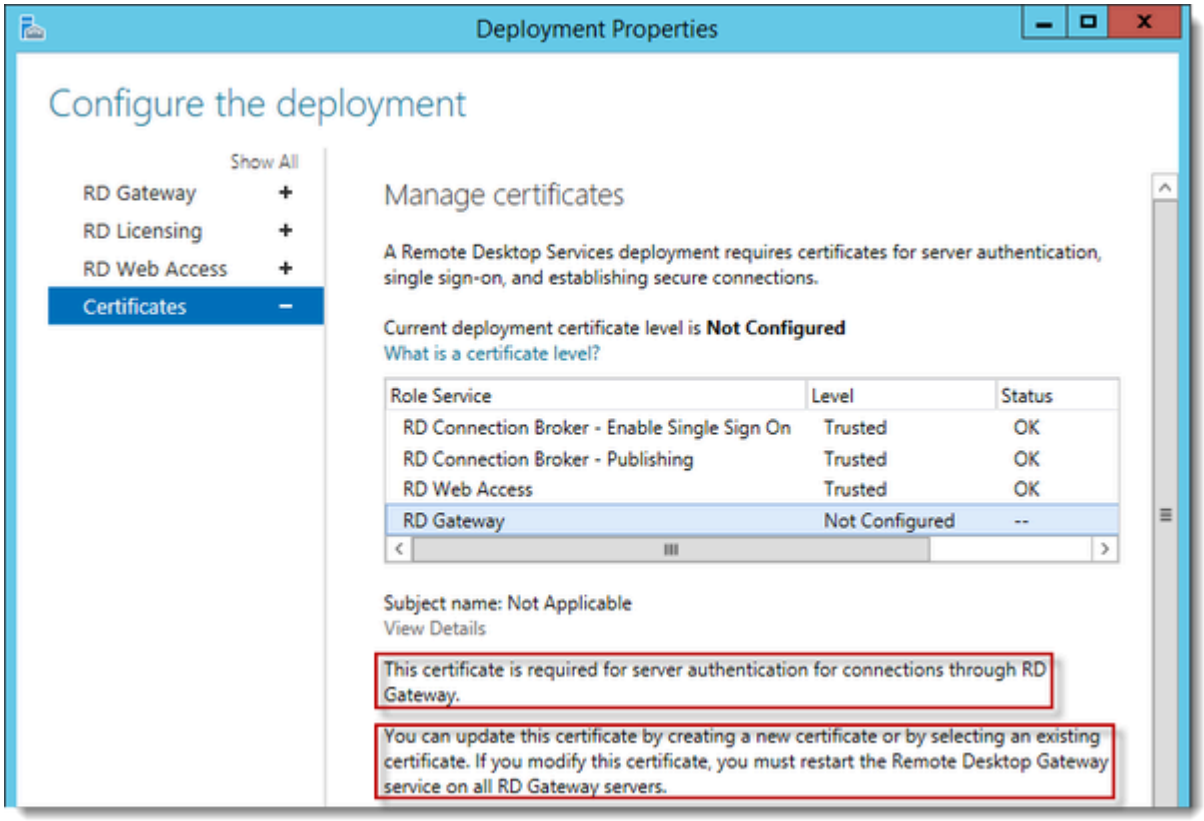
Notice the purpose of this certificate.

Click Select Existing Certificate and add the certificate you prepared for the RD Web Access server.



Click Apply to apply the certificate changes. Do not click OK because we need to configure the other certificate options as well and we can configure only one at a time.

Configure the deployment

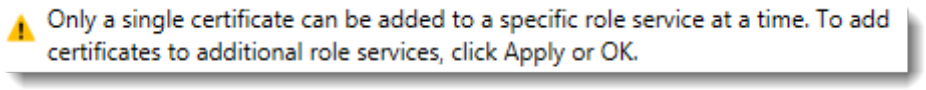


Select RD Gateway.

Notice the purpose of this certificate.

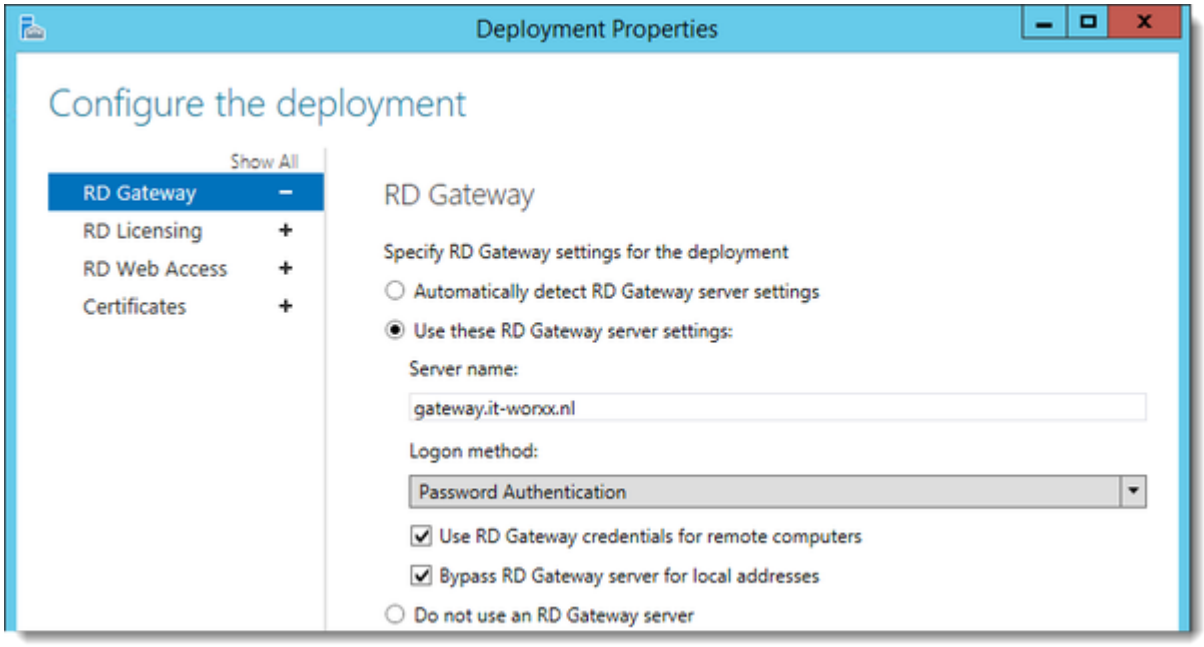
Also notice that we need to restart the RD Gateway server after we configured it to use the certificate.

Click Select Existing Certificate and add the certificate you prepared for the RD Gateway server.



Click Apply to apply the certificate changes. Do not click OK because we need to configure the rest of the deployment options, since we already have this wizard open.

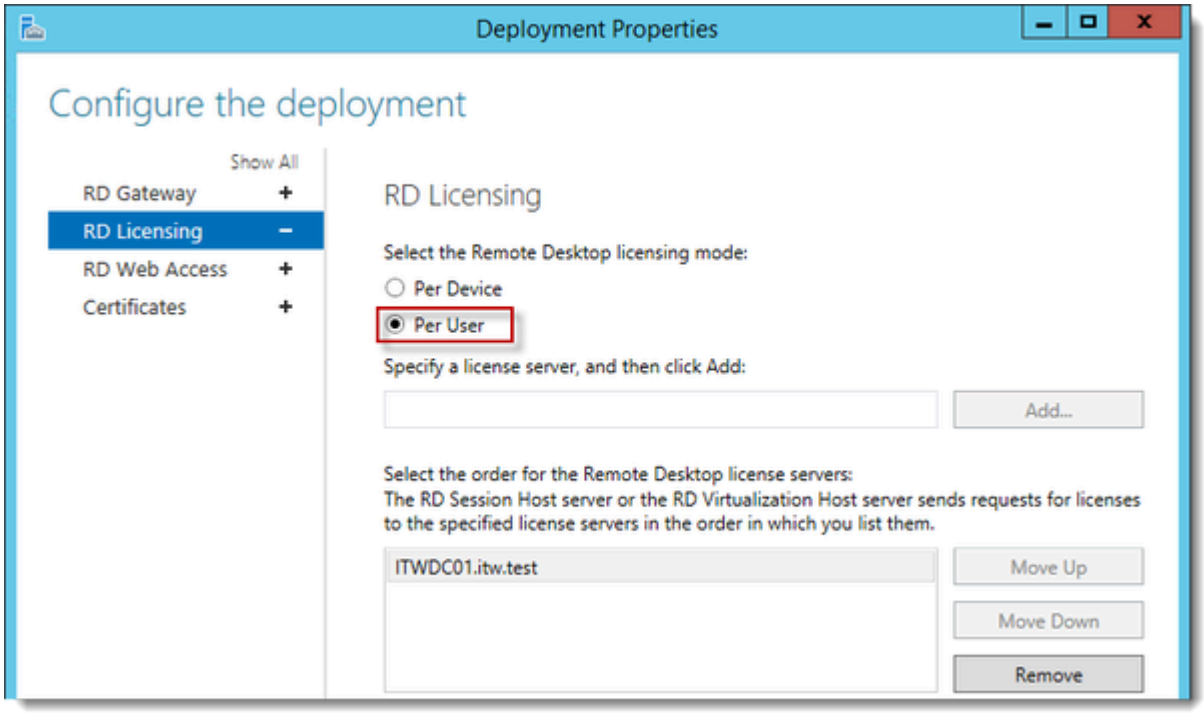
Configure the deployment



Review the RD Gateway settings and notice what settings are available.

Click RD Licensing.

Configure the deployment

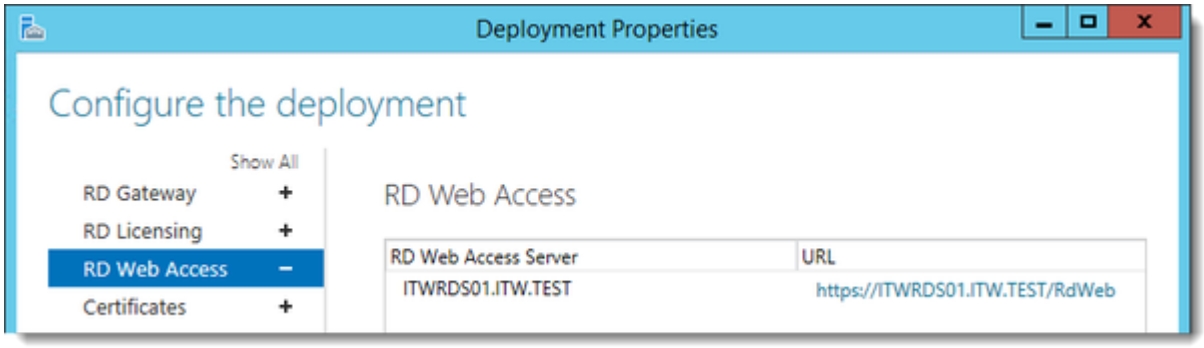


Notice that a RD License server is available, but no license type is selected yet.

I selected Per User, but since this is just a demonstration setup, it really doesn’t matter.

Click RD Web Access.

Configure the deployment



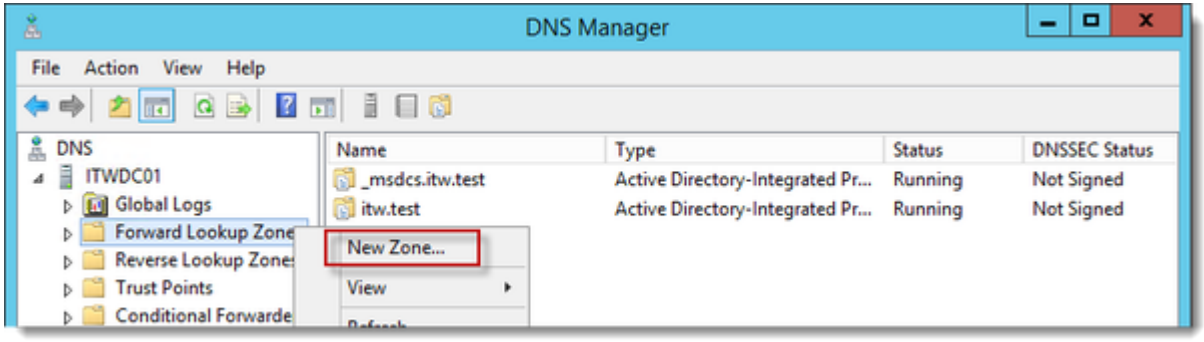
By default the RD Web Access IIS application is installed in /RdWeb. If you want to know how to change this, check another post:

https://msfreaks.wordpress.com/2013/12/07/redirect-to-the-remote-web-access-pages-rdweb/

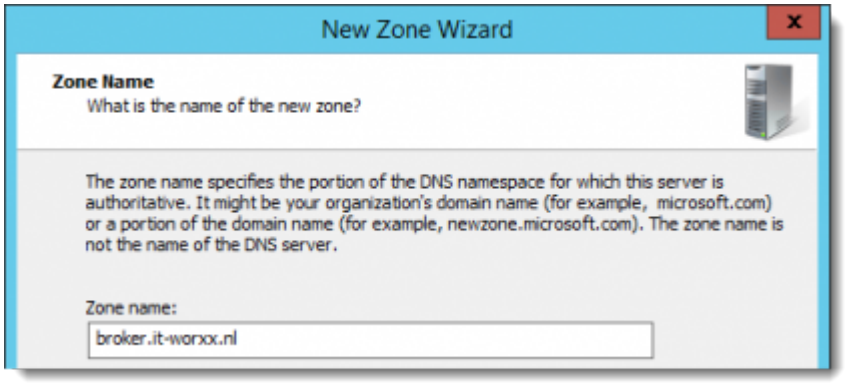
Click OK, and click Close to finish the RD Gateway wizard.

Reboot the RD Gateway server.

Open DNS Manager on the domain controller and browse to Forward Lookup Zones.



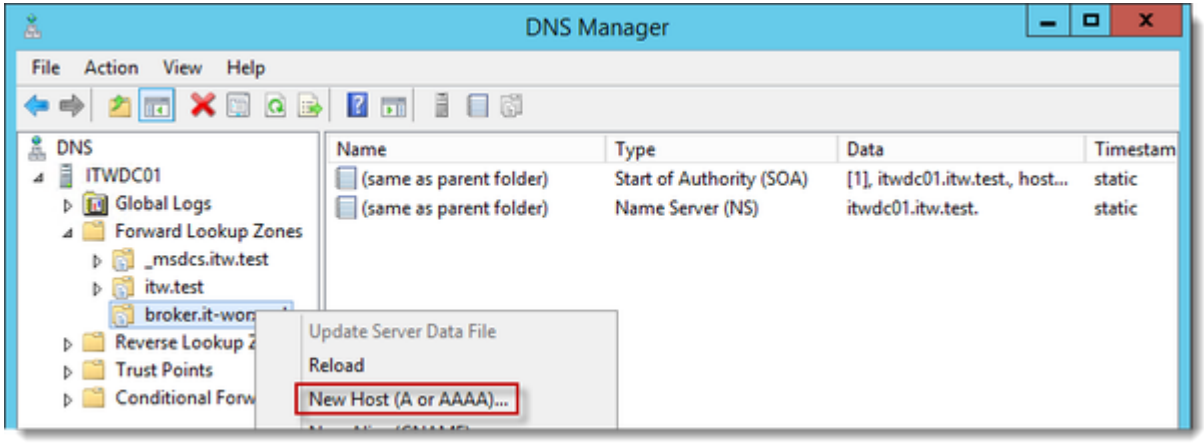
Right click Forward Lookup Zones and click New Zone… Go through this wizard accepting the defaults until you have to enter a Zone Name.



Enter the external FQDN which will also be used by the Connection Broker (which is also on the RD Connection broker's certificate.

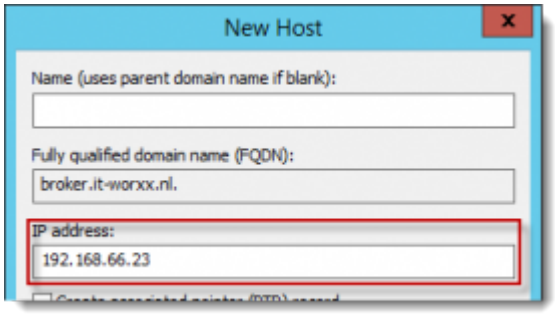
Finish the rest of the wizard accepting the defaults.

Browse to the newly created zone.



Right click the newly created zone and click New Host (A or AAAA)…

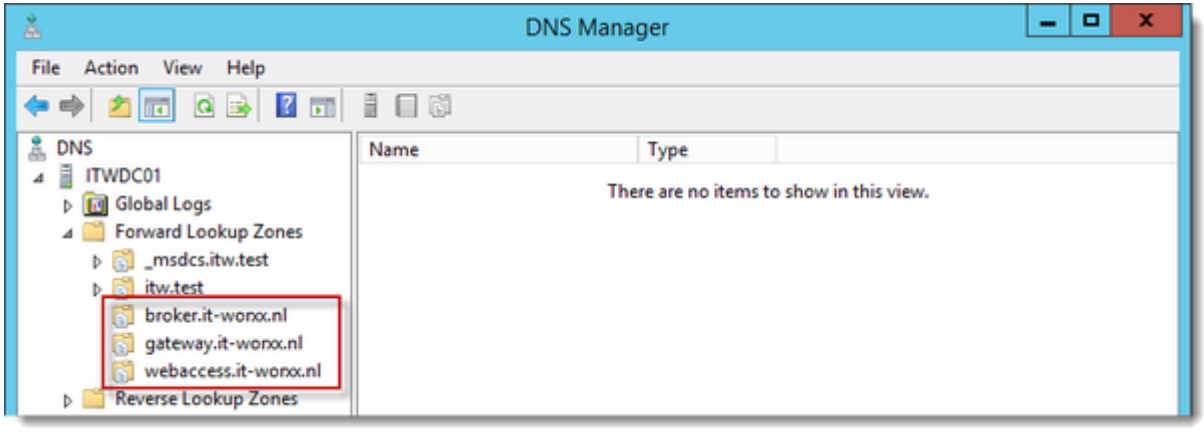
New Host



Leave the Name field blank, but enter the member server's (holding the RD Connection Broker role) internal IPv4 address.

Click Add Host.

Repeat these DNS steps for gateway.it-worxx.nl and for webaccess.it-worxx.nl.



We've effectively enabled the deployment to be useable by internal users as well by configuring these DNS zones.

Create a new Global Security Group called “RDS Connection Brokers” and add the computer account for the member server holding this role to it as a group member.

We need this group to be able to convert the RD Connection Broker to a highly available RD Connection Broker. You’ll see why we need to do this in a few steps.

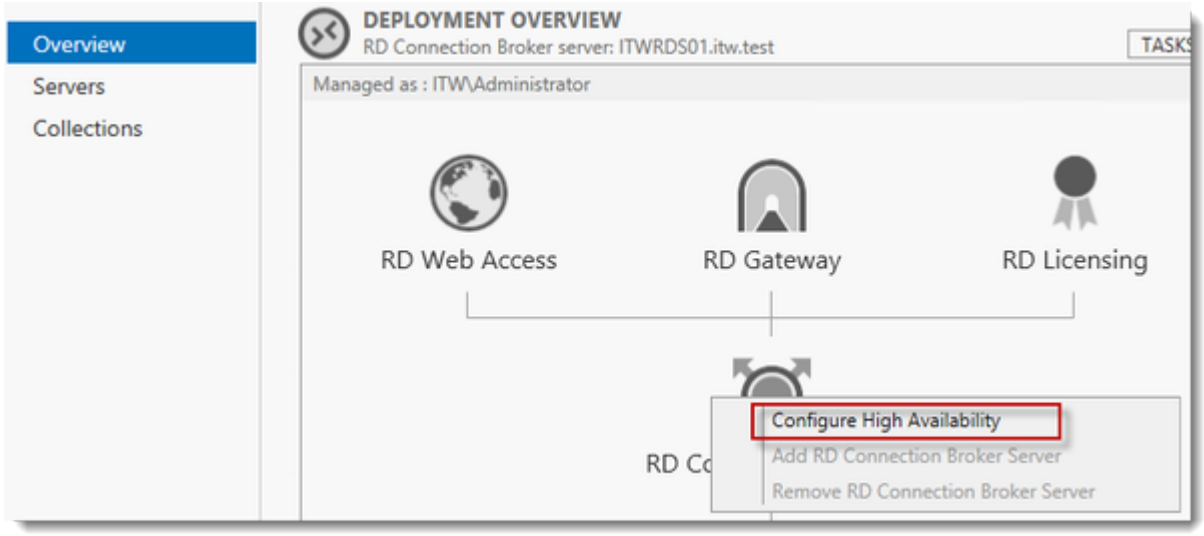
Reboot the member server holding the RD Connection Broker role to let it know it’s a member of the RDS Connection Brokers security group.

Install SQL Express on the Domain Controller (or use an existing SQL Server if you already have one). For a list of needed features, and a little more detail visit Part 1 of this series,https://msfreaks.wordpress.com/2013/12/09/windows-2012-r2-remote-desktop-services-part-1. That post lists the does and don’ts for using SQL Express with an RD deployment. This includes adding the SQL login for the RD Connection Broker servers. Do not continue with this guide unless you have a working and configured SQL environment.

Install the SQL Native Client on the member server holding the RD Connection Broker role (Client Components only). Install the client which corresponds to your SQL Server version!

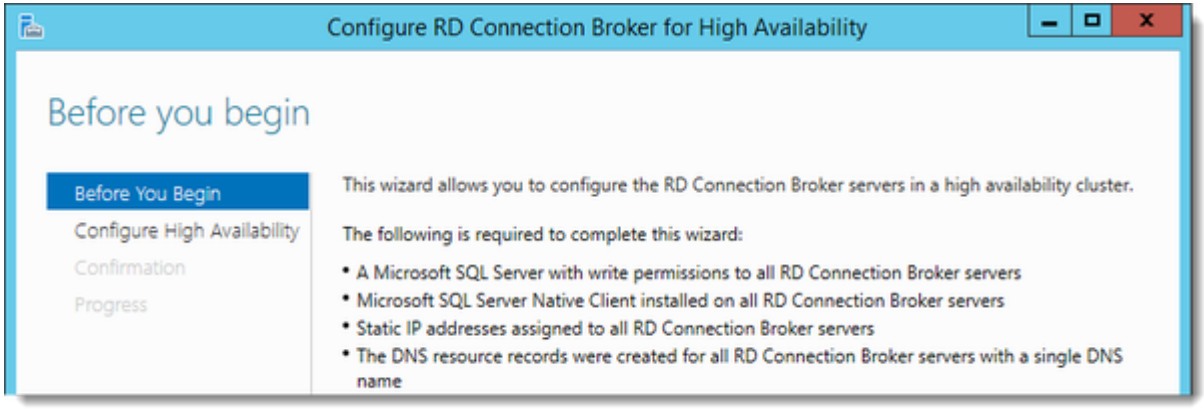
Everything we need is in place to convert the RD Connection Broker, so let’s do just that. This procedure is similar to the single server setup.

In Server Manager click Remote Desktop Services and scroll down to the overview.



Right click RD Connection Broker and click Configure High Availability.

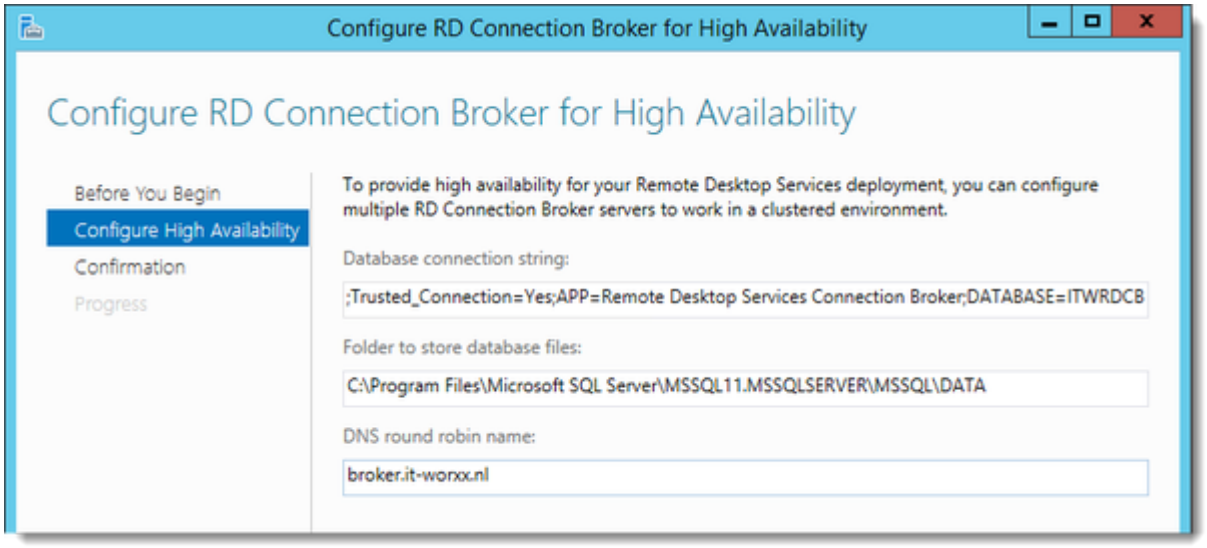
Before you begin



Look at the pre-requisites.

Click Next.

Configure RD Connection Broker for High Availability



Database connection string:

DRIVER=SQL Server Native Client 11.0;SERVER=ITWDC01;Trusted_Connection=Yes;APP=Remote Desktop Services Connection Broker;DATABASE=ITWRDCB

- Or any other database name you want, the database will be created by this wizard.
- Replace the DRIVER= part with the version you installed if it's anything other than SQL Server 2012 (SP1)

Folder to store database files:

C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA

I used the instance default folder.

- Note that this points to a folder on the SQL Server.

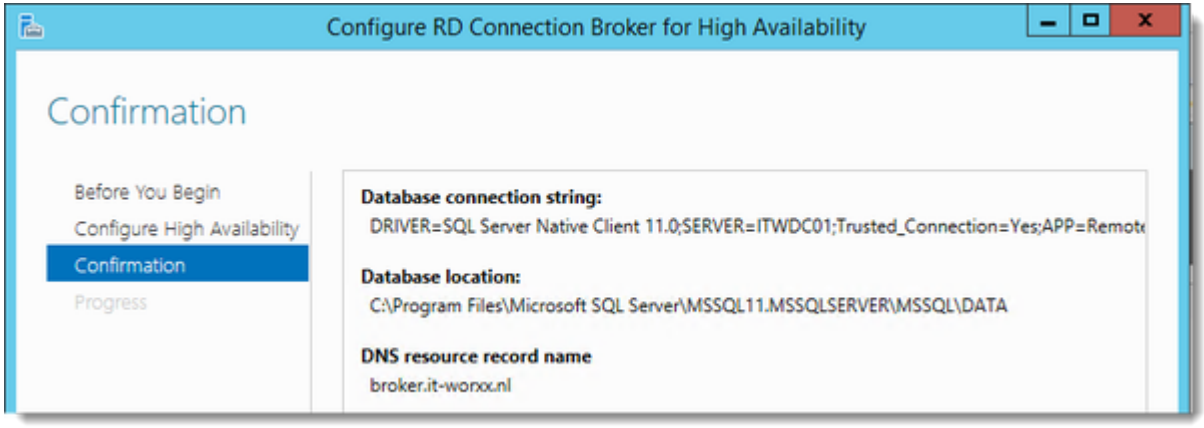
DNS round robin name:

The DNS Zone name we configured in DNS earlier.

- And now you see why we had to create this zone in internal DNS as well. This needs to be locally resolvable.

Click Next.

Confirmation

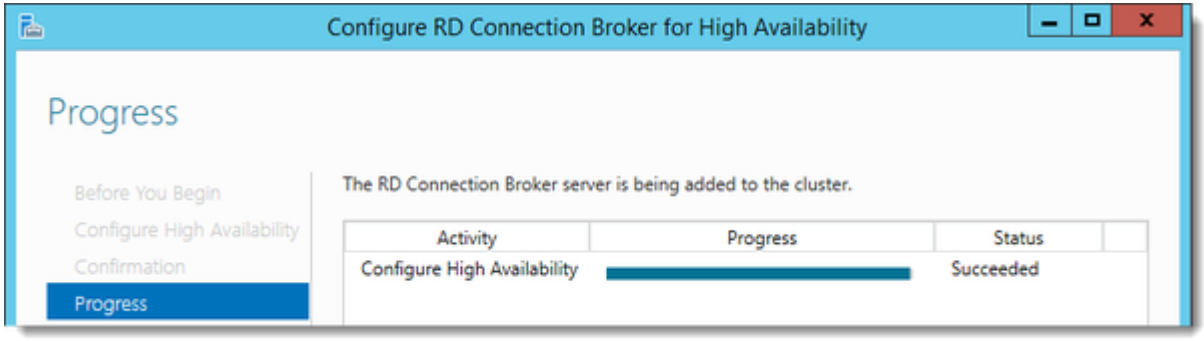


If you get an error before this page:

- Check if TCP/IP is enabled in client protocols and for your instance
- Check if you can reach port 1433 on the SQL Server from the member server

Click Configure.

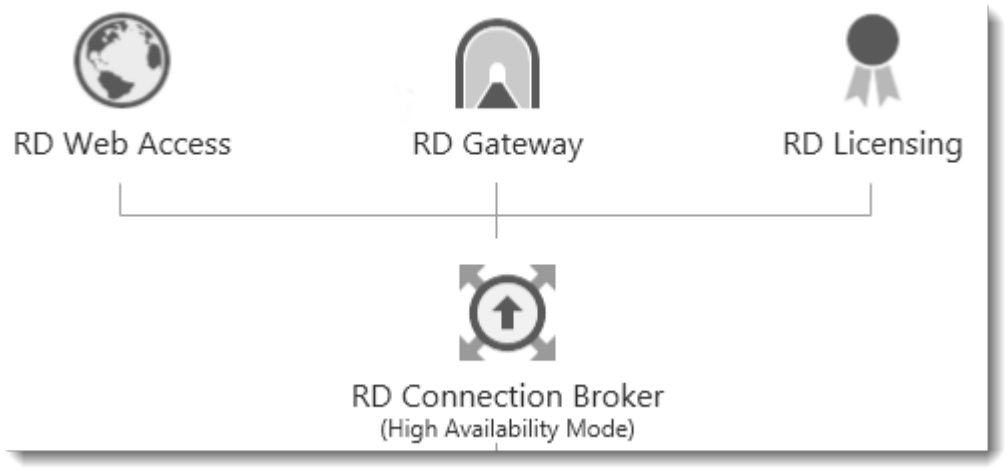
Progress



If you get an error on this page:

- Check SQL permissions for the security group
- Check if the database path you entered is correct

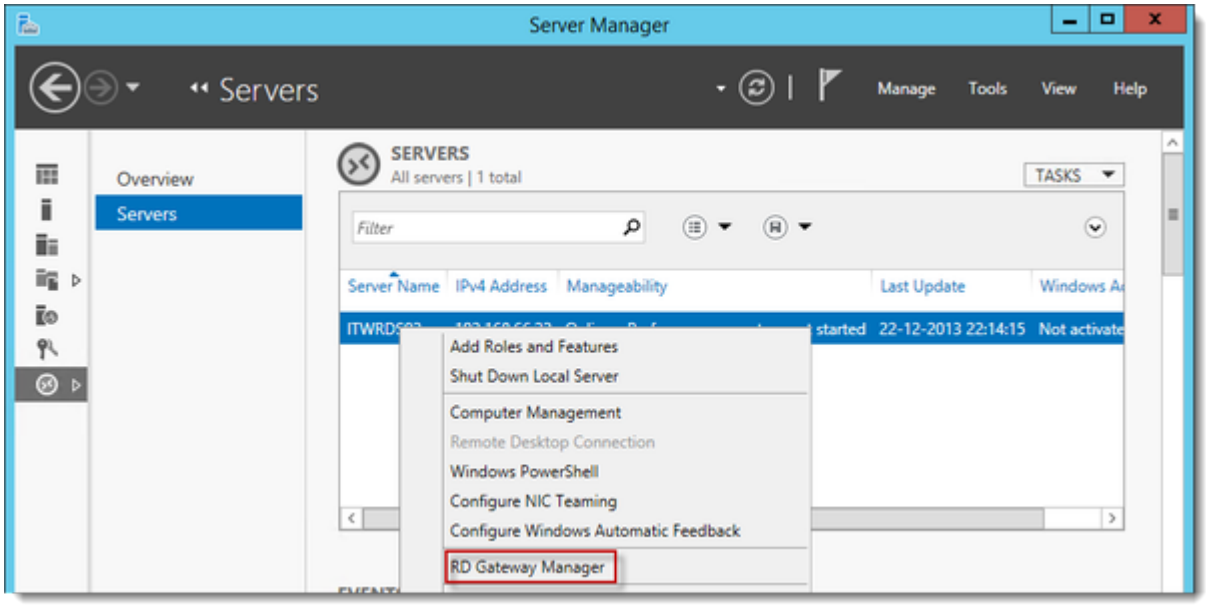
Click Close.



The RD Connection Broker is now in High Availability Mode and we are finally ready to complete the configuration.

Since the RD Connection Broker is known within the deployment for broker.it-worxx.nl and thus not a FQDN that's associated with the internal domain (itw.test) we need to tell the gateway that external users are allowed to connect to it.

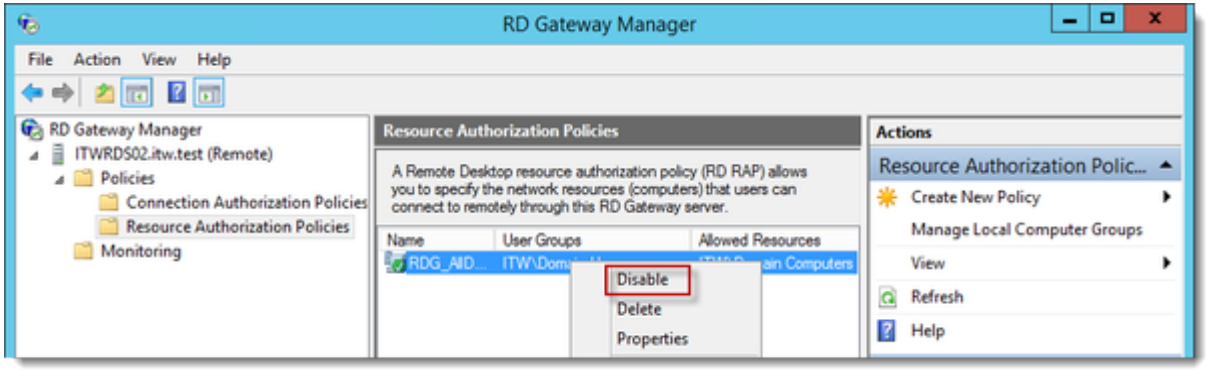
On the RD Gateway server, open Server Manager



Click Remote Desktop Services (yes, it says it's missing servers, just ignore this), click Servers and then right click the RD Gateway server.

Click RD Gateway Manager.

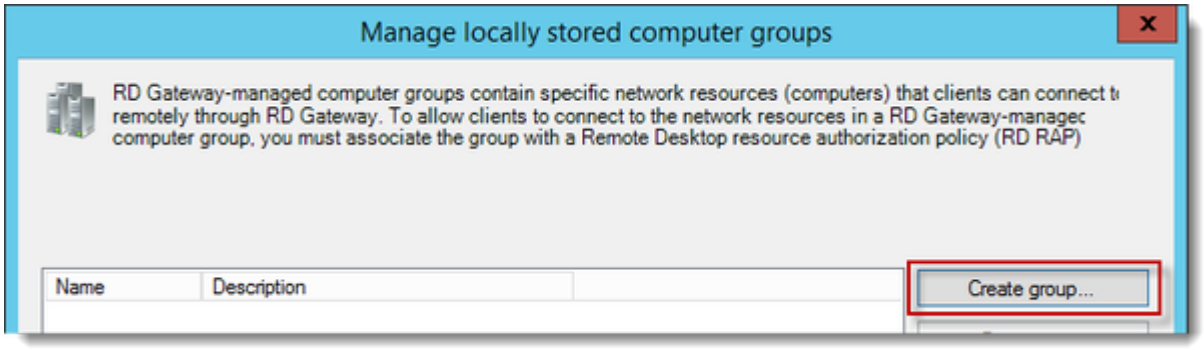
RD Gateway Manager



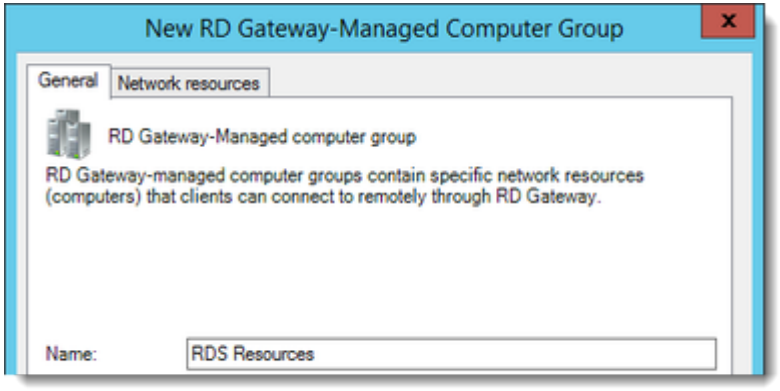
Navigate to Policies – Resource Authorization Policies. There's the default policy. Right click the default policy and disable it.

In the Actions pane to the right, click Manage Local Computer Groups.

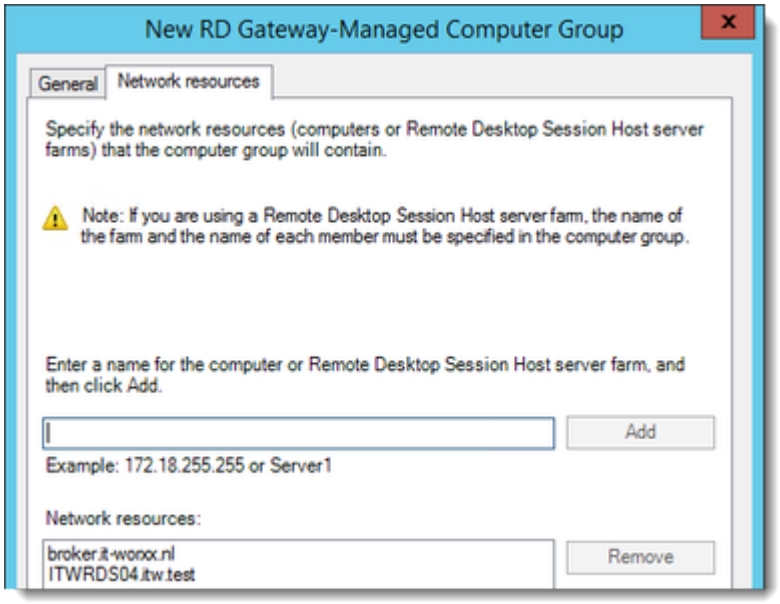
Manage locally stored computer groups



Click Create group…



Name the new group.

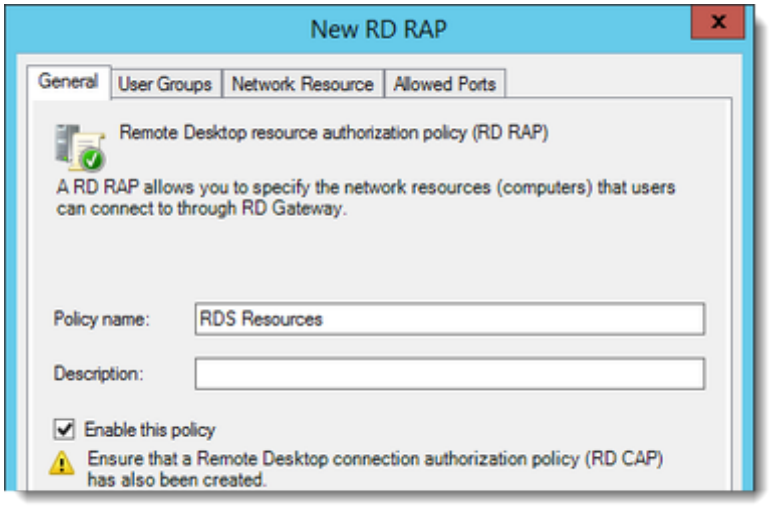


On the Network Resources tab, add the RD Session Host(s) and the DNS external name of the broker.

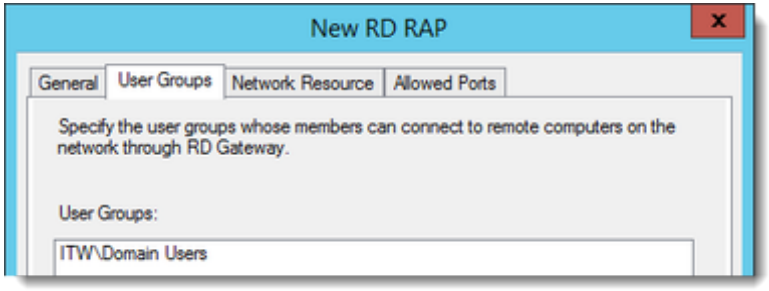
Click OK.

RD Gateway Manager

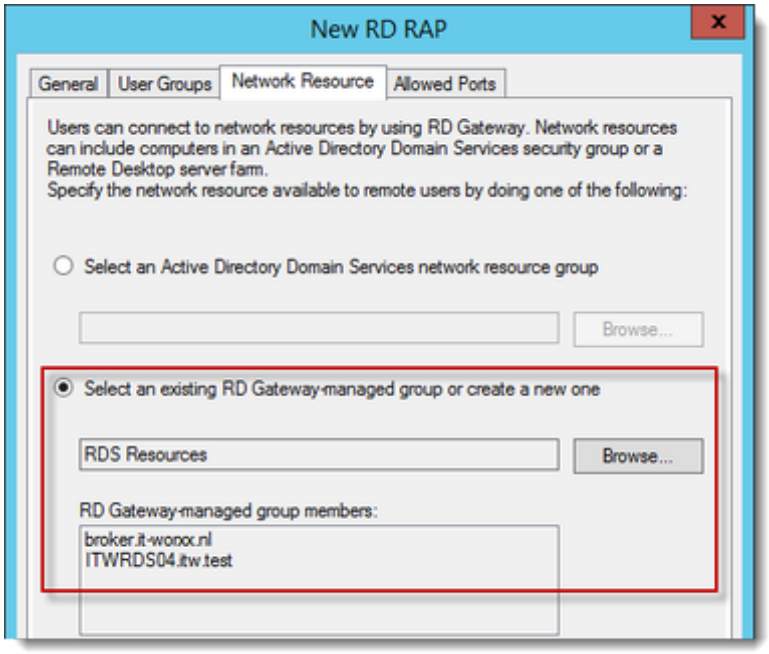
Right click the Resource Authorization Policies node, click Create New Policy, Click Custom.



Name the policy, click User Groups



Add Domain Users, or any group you wish to grant access, click Network Resource



Click Select an existing RD Gateway-managed group or create a new one, and then browse to select the group you created a few steps back. Notice that upon selecting the group the RD Gateway-managed group members box shows the members of the group.

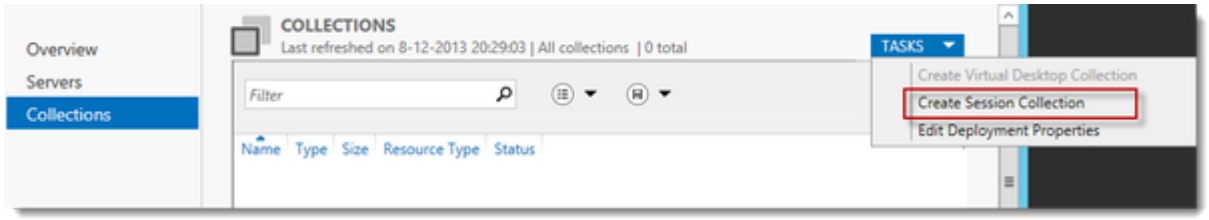
Review the Allowed Ports tab.

Click OK.

That’s it, configured all servers, configured certificates, configured RAP..

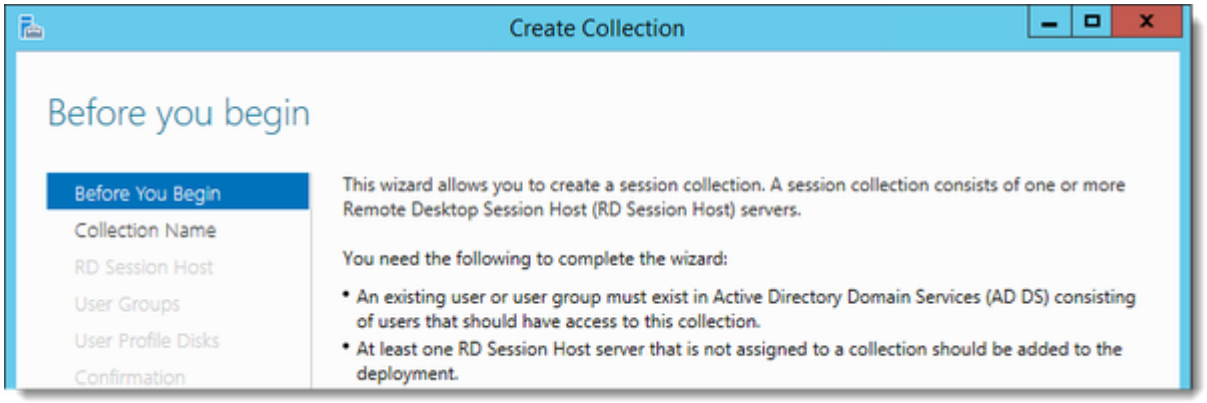
One thing left to do: Tell our RDS environment exactly what to publish.

Let’s publish full desktop sessions again, like in the single server setup. Next post we we’ll dig into publishing remote applications, I promise :)



In Server Manager, Remote Desktop Services, Session Collections, click Tasks and click Create Session Collection.

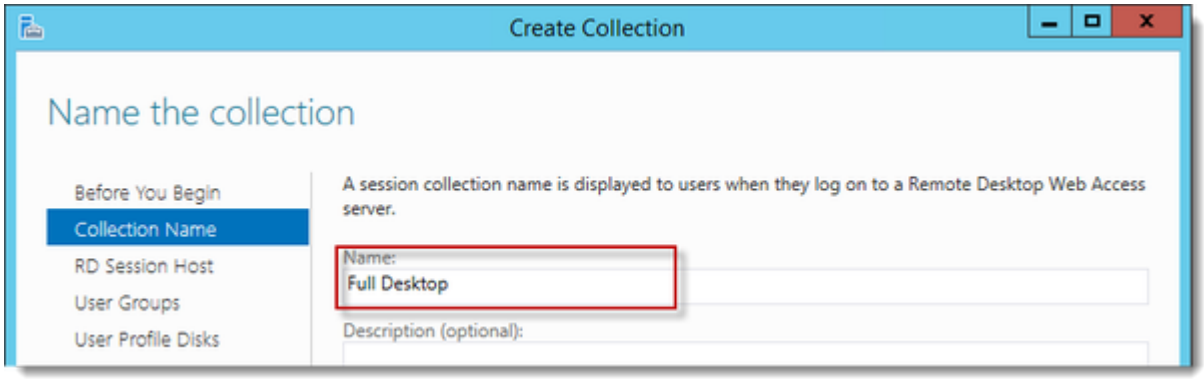
Before you begin



Review the requirements. This won't be an issue in this setup, but you could restrict access to this collection by selecting a select group of people.

Click Next.

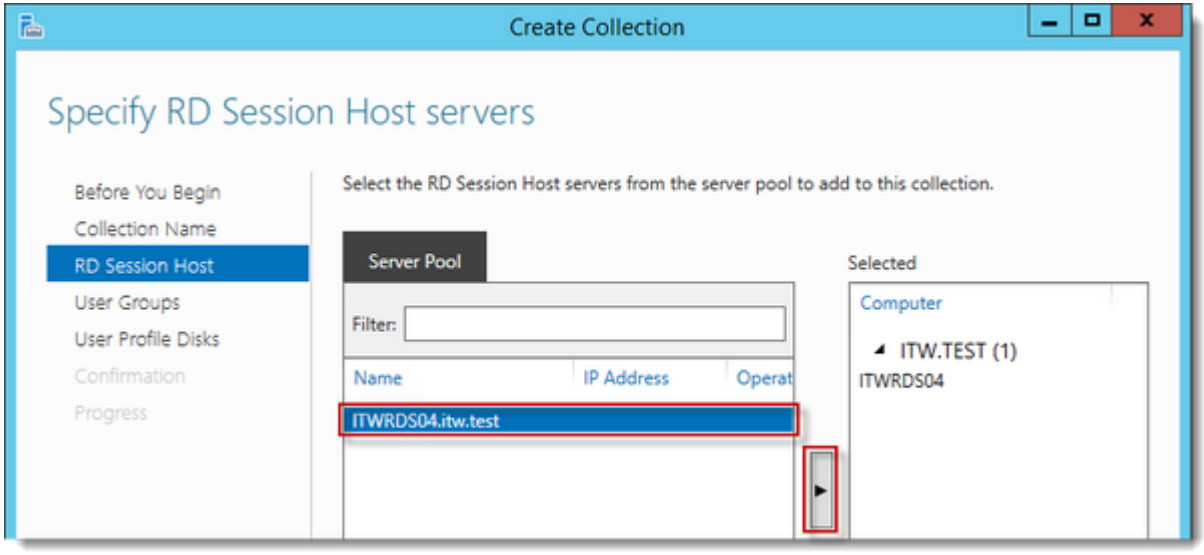
Name the collection



Enter a descriptive name. This name will be displayed under its icon in the Web Access interface.

Click Next.

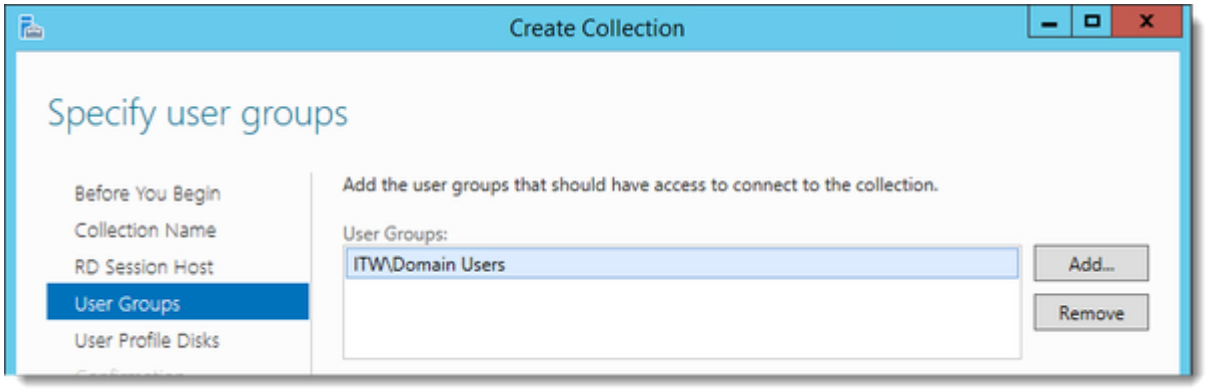
Specify RD Session Host servers



Click the member server holding the RD Session Host role and click the Add button.

Click Next.

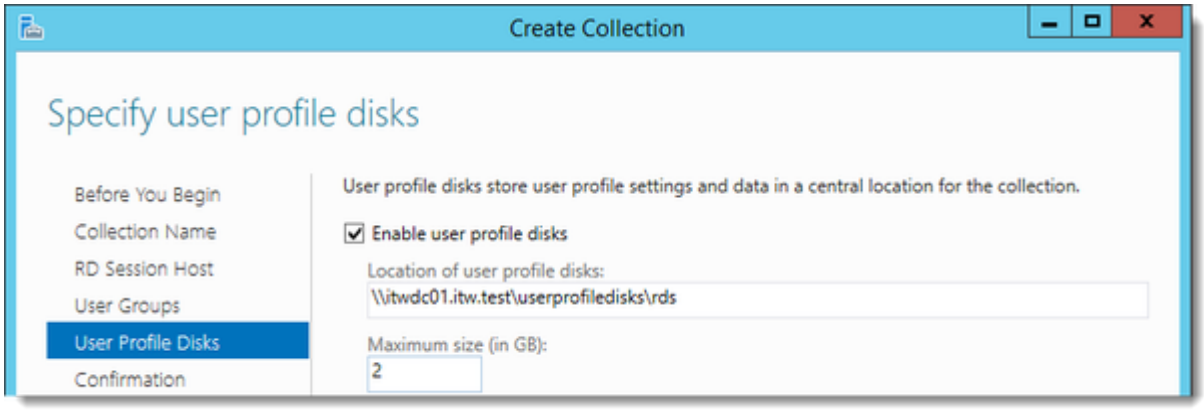
Specify user groups



You can limit access here. Add one or more groups to restrict access to these groups only. In this setup Domain Users will do fine.

Click Next.

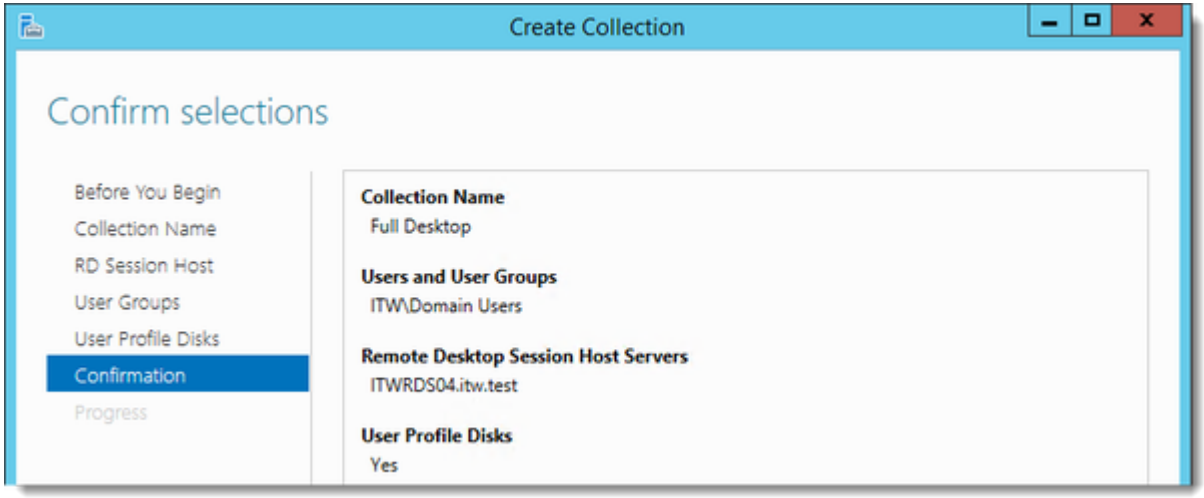
Specify user profile disks



First, create a folder on the domain controller “UserProfileDisks” and a subfolder “RDS”. Share “UserProfileDisks”. Now in the Create Collection wizard enter \\itwdc01.itw.test\userprofiledisks\rds and set the Maximum size to 2GB. Further does and don’ts for User Profile Disks will be covered in a future post.

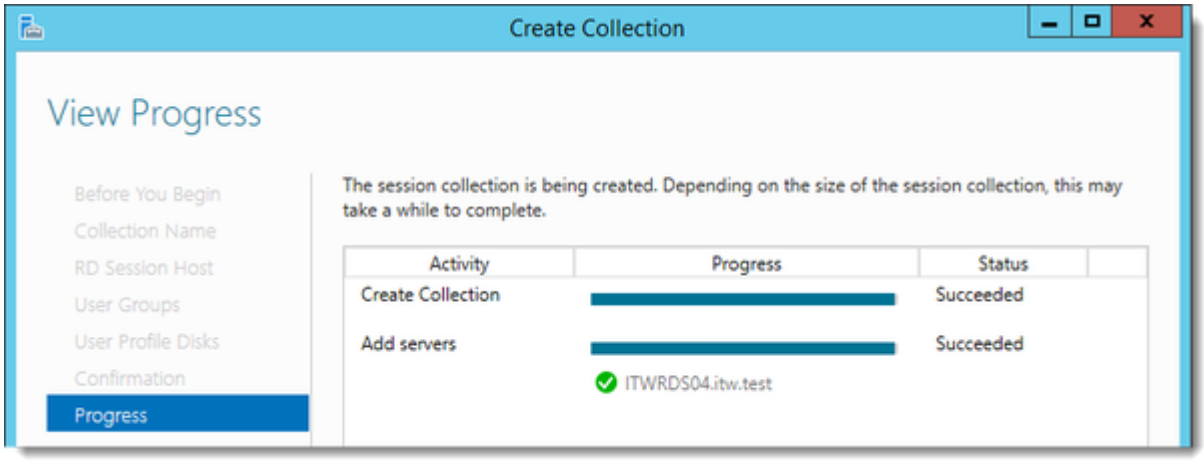
Click Next.

Confirm selections



Review the information and click Create.

View Progress

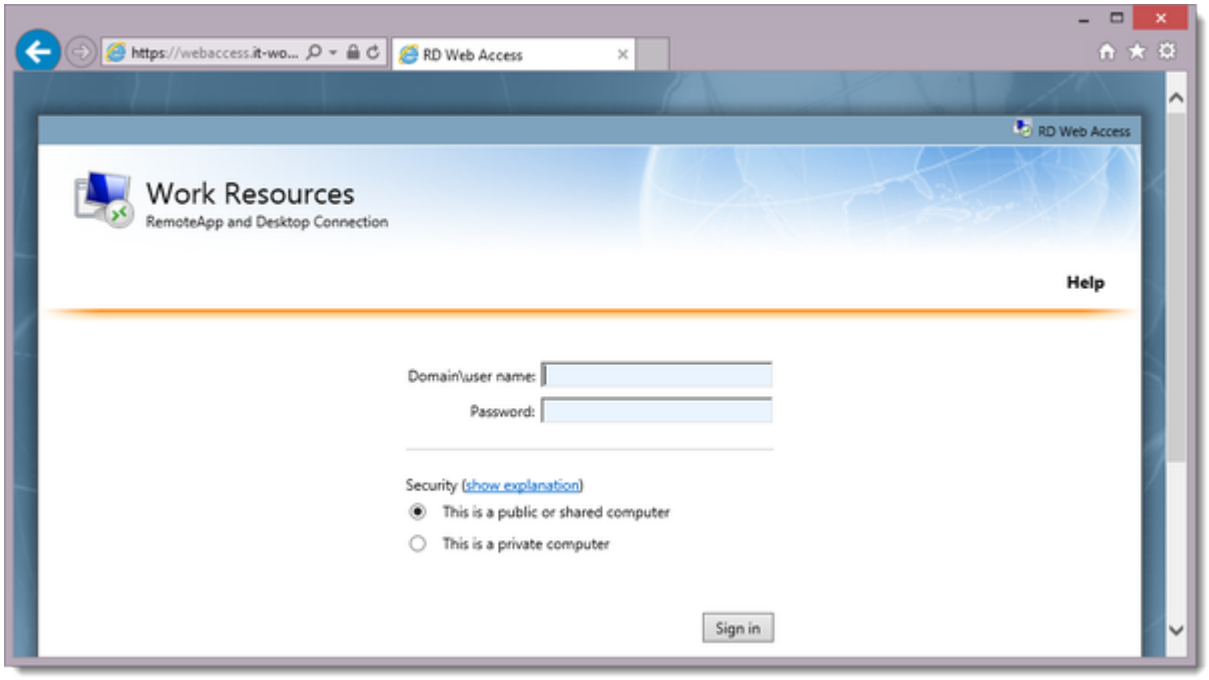


Wait until the collection is created and the server is added to the collection.

Click Close.

Time to test the setup!

On a machine that has access to your test setup (you may have to add the external FQDN for the RD Gateway and for the RD Web Access to your hosts file if you didn't publish it to the internet) open https://webaccess.it-worxx.nl/rdweb



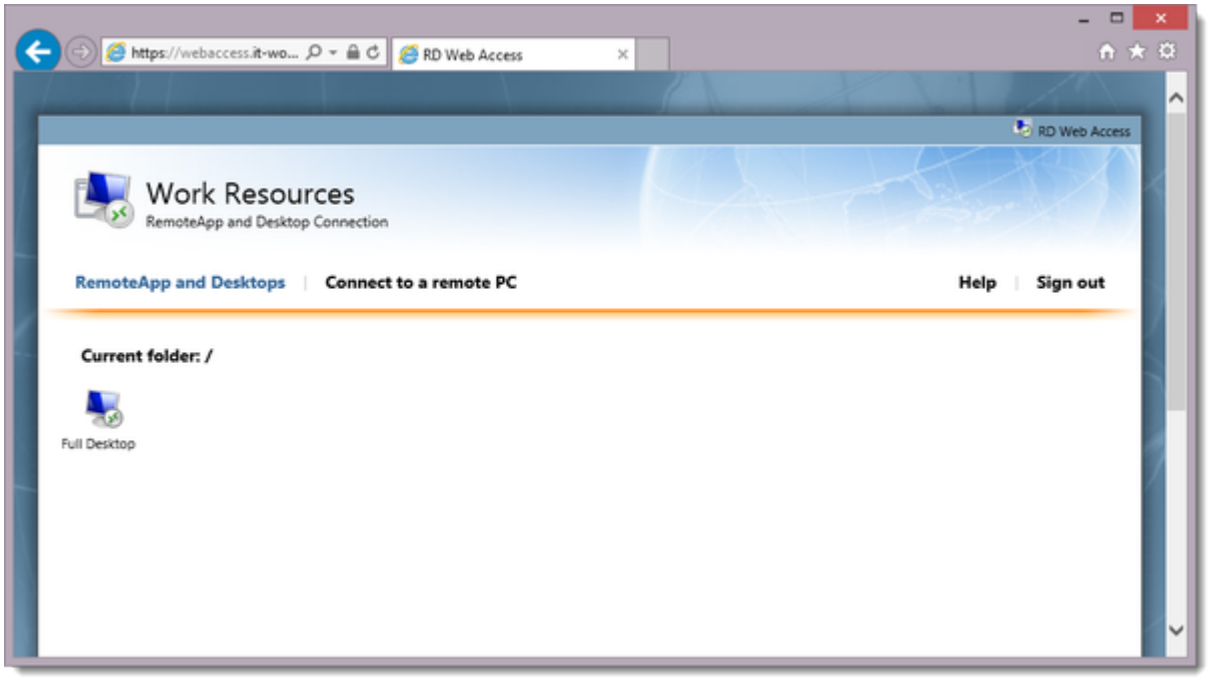
Hey! The RD Web Access application works. If you want to get rid of the /RDWeb part in the url, check out this post:

https://msfreaks.wordpress.com/2013/12/07/redirect-to-the-remote-web-access-pages-rdweb

Enter a valid username and password (ITW\username or username@itw.test).

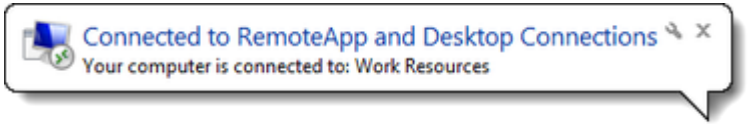
Create a user for this, or simply use the domain admin account.

Click Sign in.



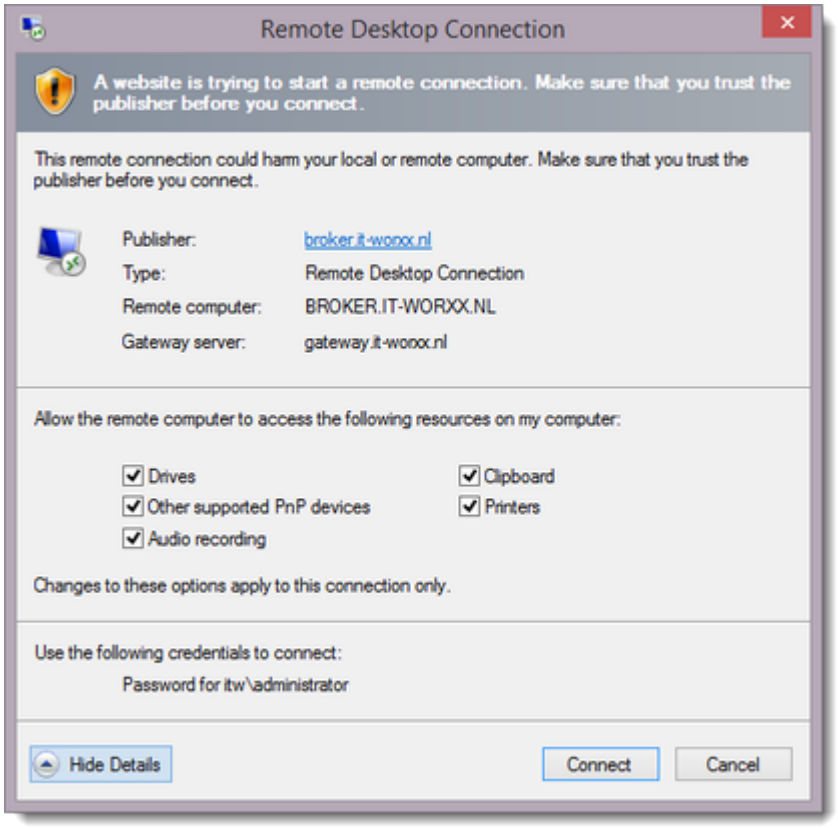
After logging in you are presented with the full desktop session collection we created.

Also notice the popup in your taskbar as soon as you're connected:



Again, sorry, but I'll handle that in a future post.

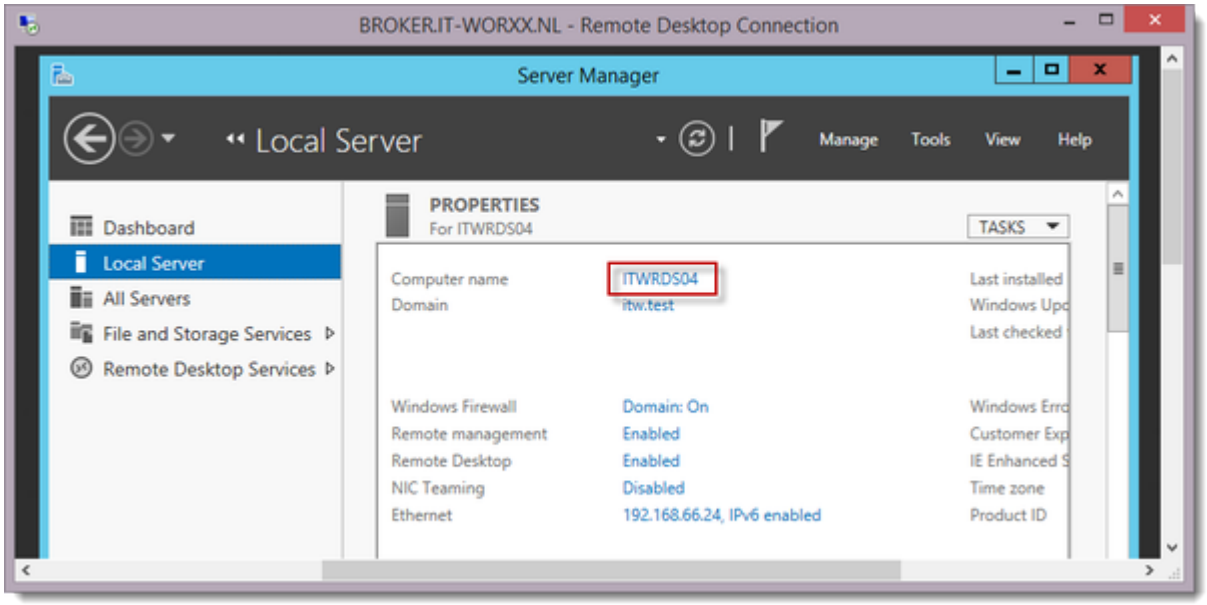
Click the "Full Desktop" icon to open it and another popup appears:



This is just a warning that the resource you're requesting wants to redirect your local devices.

But it also tells us that it is signed by "broker.it-worxx.nl", and we're using a gateway to connect to the remote resource..

And when you click Connect, you actually connect.



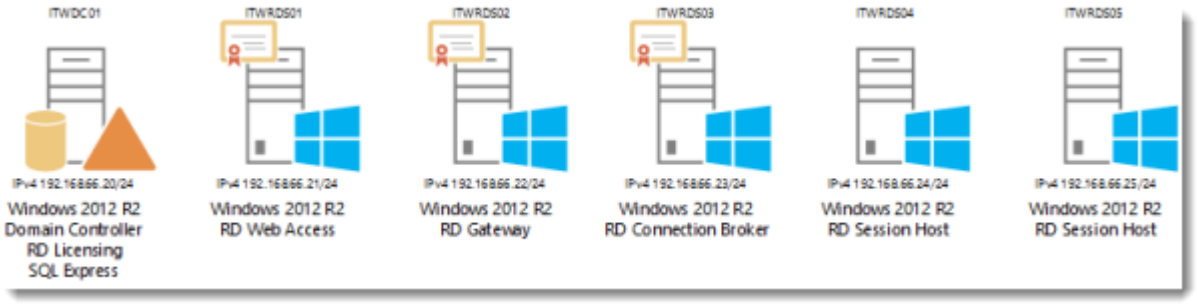
Because I connected as an admin I can see on which server I am logged on by clicking Local Server. And this screenshot also shows that it's the broker that provided me the connection..

In the next part of this series I will show how to extend this setup with another RD Session Host, but this time we'll publish some apps. Oh, and that post will probably be a lot shorter.

Step by Step Windows 2012 R2 Remote Desktop Services – Part 3

A step by step guide to build a Windows 2012 R2 Remote Desktop Services deployment.
Part 3 – Adding Session Hosts and Load Balancing session collections.

In this step by step guide we'll be adding an extra RD Session Host server:



ITWRDS05 will be the extra server. I used the same specs as in step 2 in this guide for the member servers, and used IPv4 192.168.66.25/24 and made it a member server of the domain.

If you're building along and want to continue doing so for the next parts in this complete series, make snapshots of the servers before adding this extra server.

Software used in this guide:

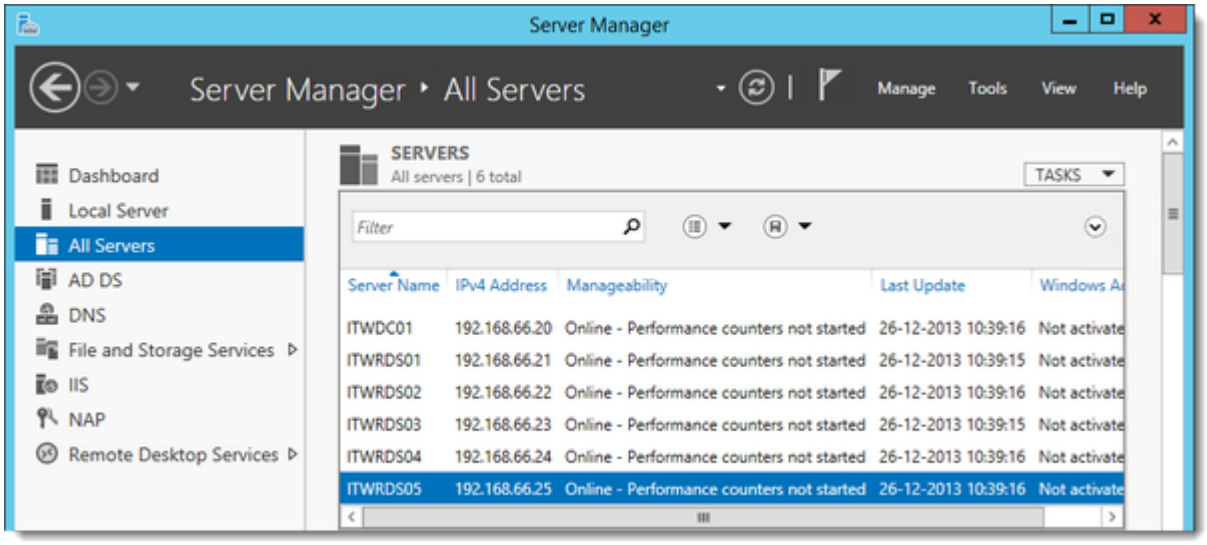
Windows Server 2012 R2 ISO (evaluation can be downloaded here:http://technet.microsoft.com/en-us/evalcenter/dn205286.aspx)

This guide will not focus on adding a member server to the domain.

And again some basic knowledge is assumed in this guide.

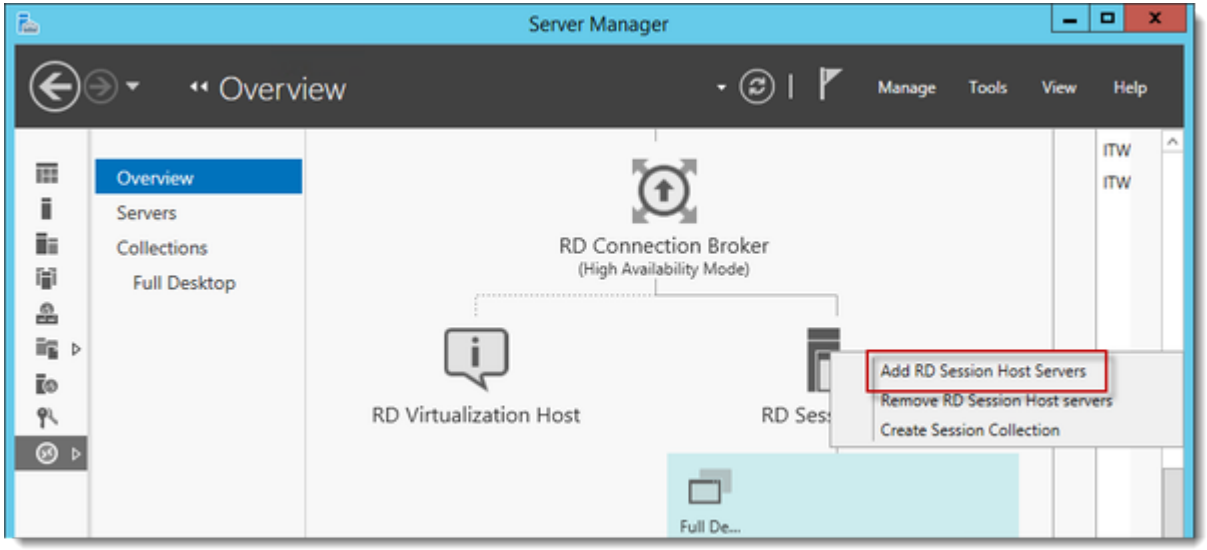
Installing the Remote Desktop Services Roles

Log on to the Domain Controller, and in Server Manager right-click the All Servers node and add the new server using the Add Servers command (or select the All Servers node, click Manage and click Add Servers).



Now that all servers needed in this deployment scenario are present, click Remote Desktop Services.

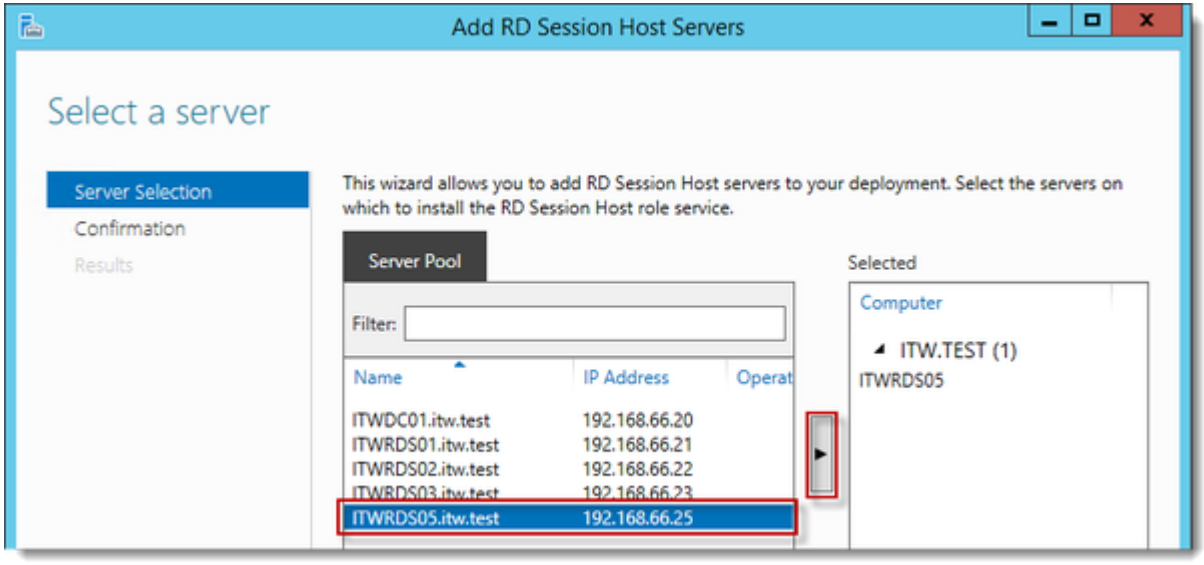
Server Manager



In Overview, right-click RD Session Host and click Add RD Session Host Servers.

Note that the Remove RD Session Host servers option is used to remove one or more Session Host servers from the deployment. This will not uninstall the RD Session Host role service from the selected server(s), unless you choose to do so in the wizard.

Select a server

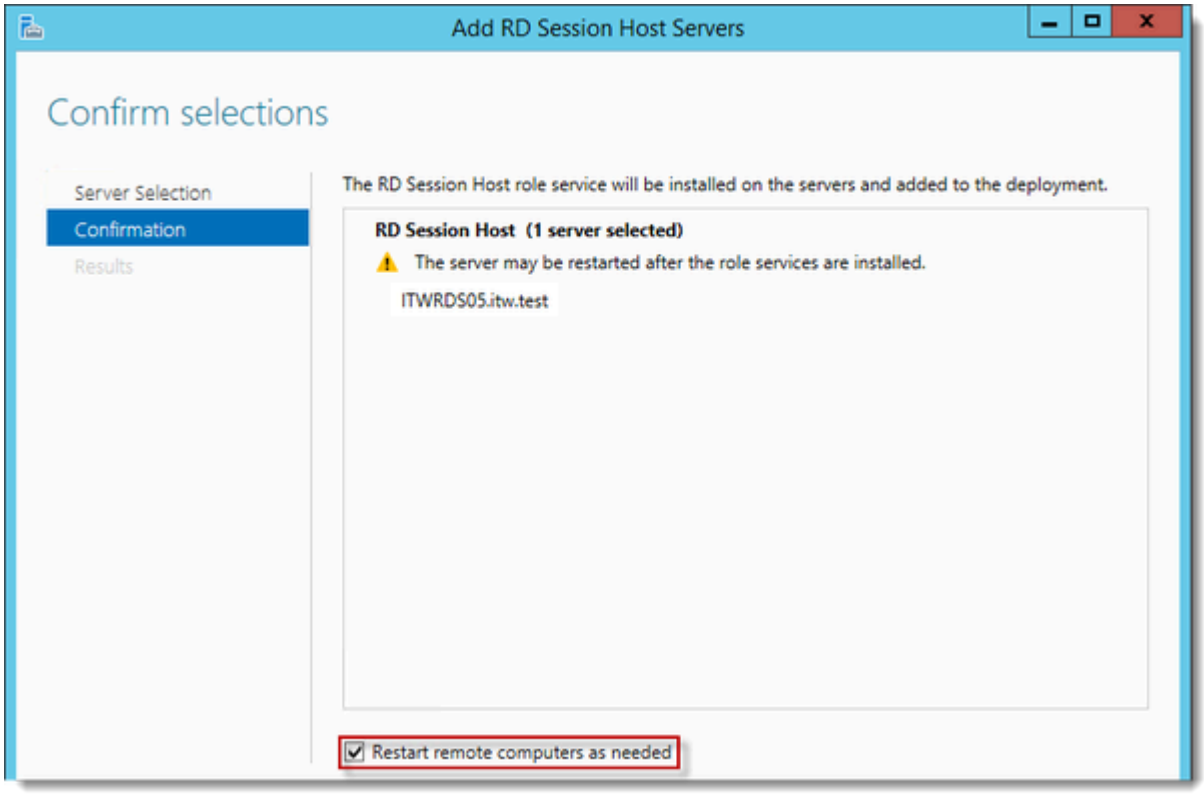


Click the newly added server and click the Add button.

Notice here that the only server missing to choose from is ITWRDS04, which is of course because this already is a RD Session Host in the current deployment.

Click Next.

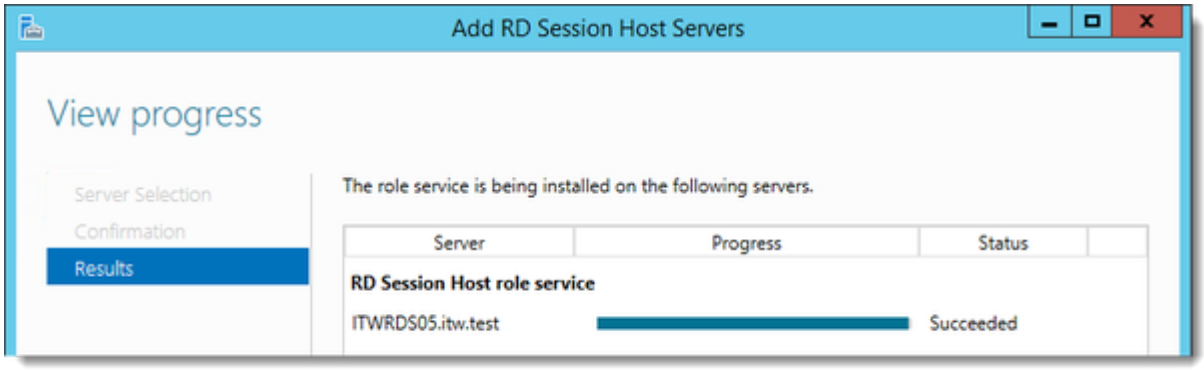
Confirm selections



Check Restart the destination server automatically if required.

Click Add.

View progress



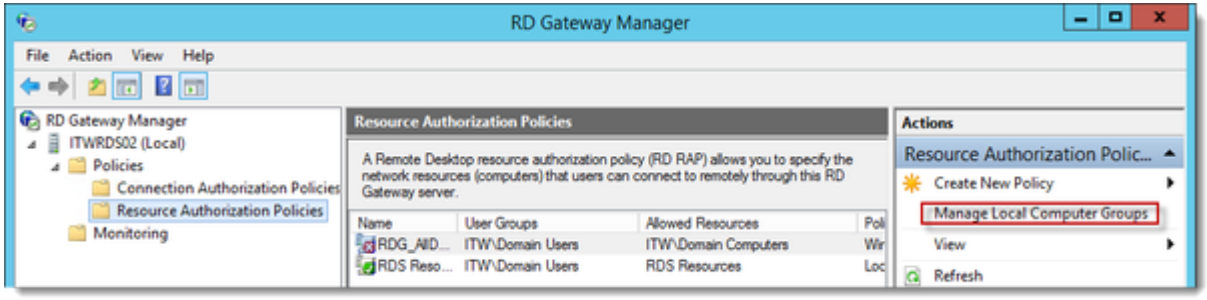
Wait until the RD Session Host role service is deployed and the new RD Session Host server has restarted.

Click Close.

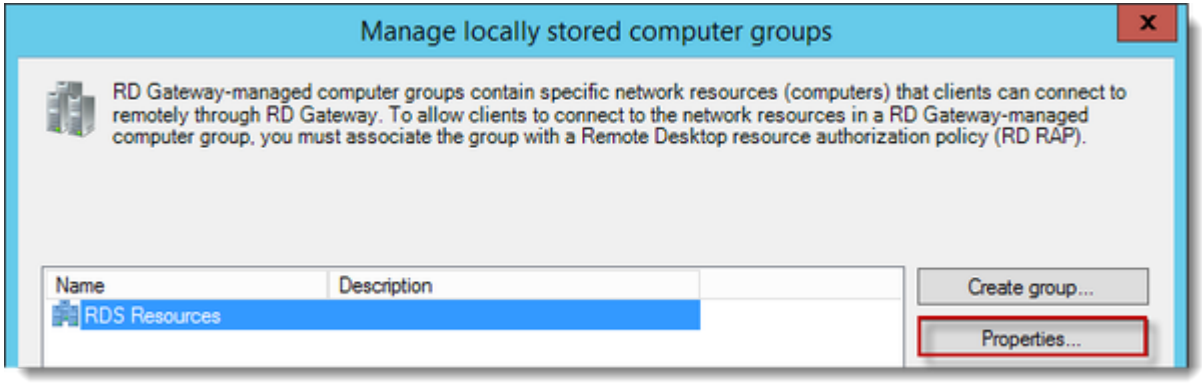
If you want Web Access users to be able to log on to this server, you need to add this server to the Resource Group for which we configured a policy on the RD Gateway server in the previous guide.

On the RD Gateway server, open the RD Gateway Manager tool and expand the server node, expand the Polices node and click the Resource Authorization Policies node.

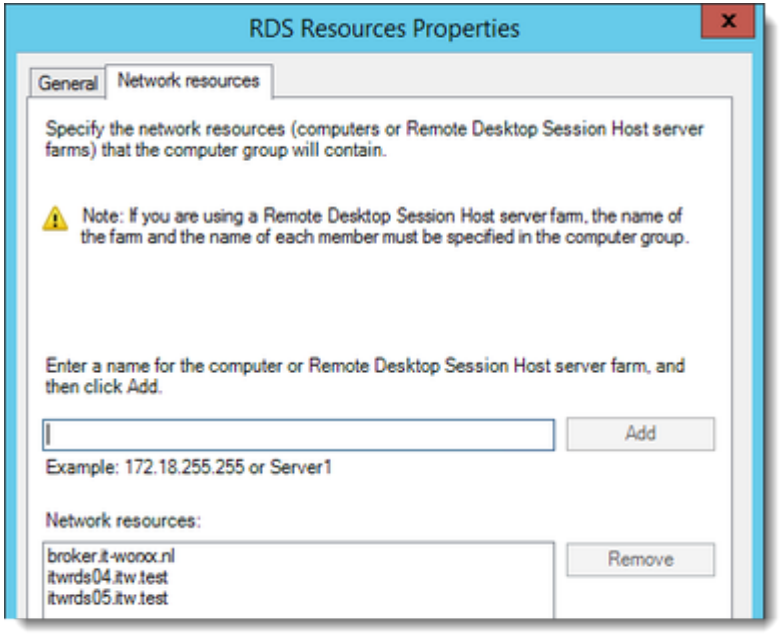
RD Gateway Manager



Click Manage Local Computer Groups.



Make sure the Resource group is selected and click Properties.



Type the name of the new server and click Add.

The Note you see here refers to the Remote Desktop Session Host server farm principle in case you also publish Windows 2008(R2) Remote Desktop deployments. In Windows 2012(R2) the farm concept is handled by the RD Broker and the RD Session Collections.

Click OK to apply the settings to the resource group and click Close to close the group manager.

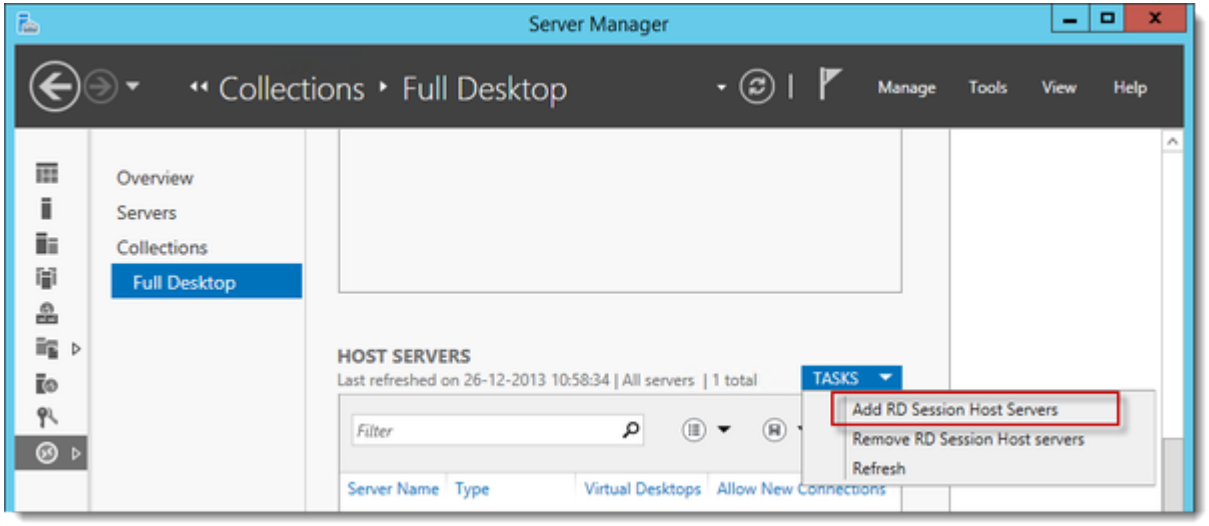
Now let's see what we can do if we have multiple Session Hosts in our deployment.

Of course you could add a new collection using the new session host server, but that's no different than what I explained in step 2 of this guide collection.

Let's do some new stuff with the new session host instead.

Load balancing an existing collection

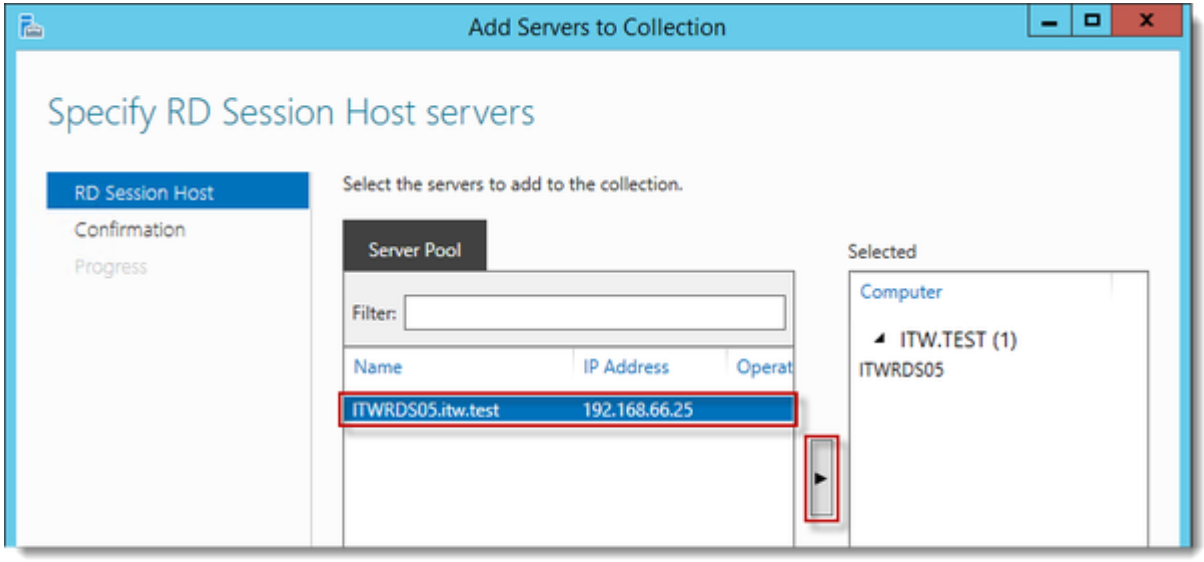
In Server Manager click Remote Desktop Services, and then click the existing collection "Full Desktop". Scroll down to Host Servers if this section is not immediately visible.



Click Tasks and click Add RD Session Host Servers.

Note that the Remove RD Session Host servers option is used to remove one or more servers from a load balanced session collection.

Specify RD Session Host servers

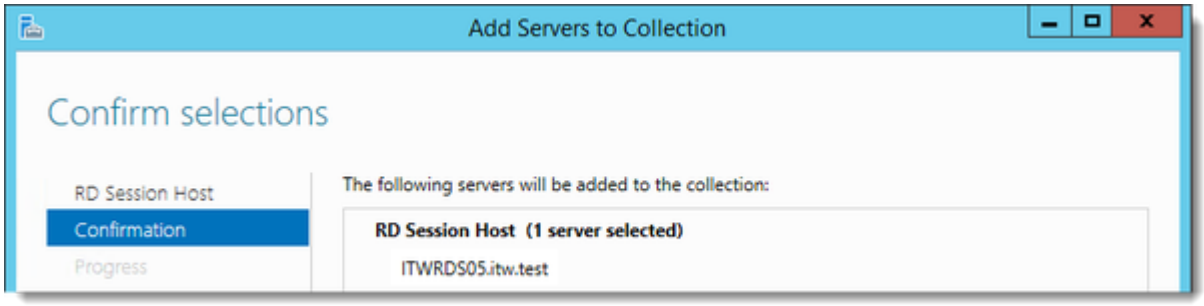


Since there's only the new server in the deployment which has the role but isn't assigned yet, that's the only server we see here.

Select the server and click the add button.

Click Next.

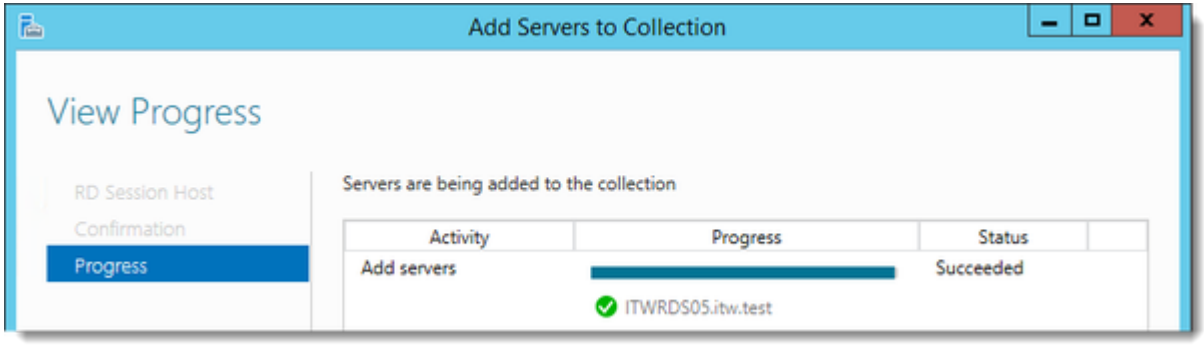
Confirm selections



The Wizard confirms that you selected the server.

Click Add.

View progress



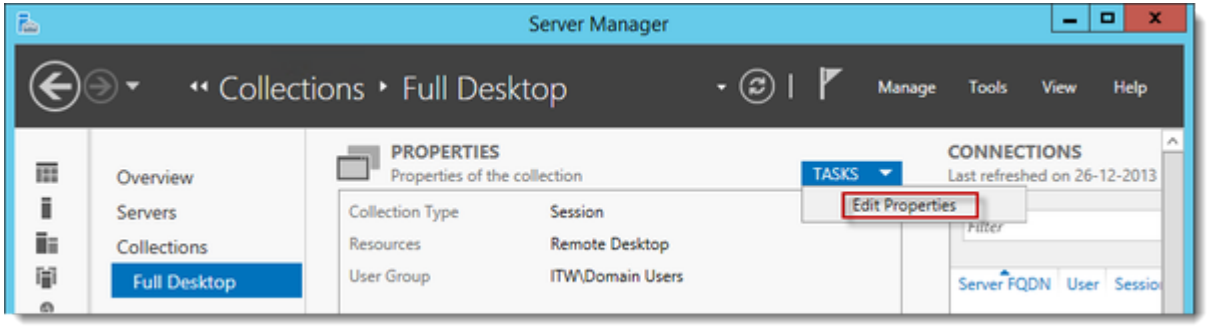
Wait until the server is added to the collection.

Click Close.

That’s it. The Full Desktop collection is now load balanced over 2 Session Hosts.

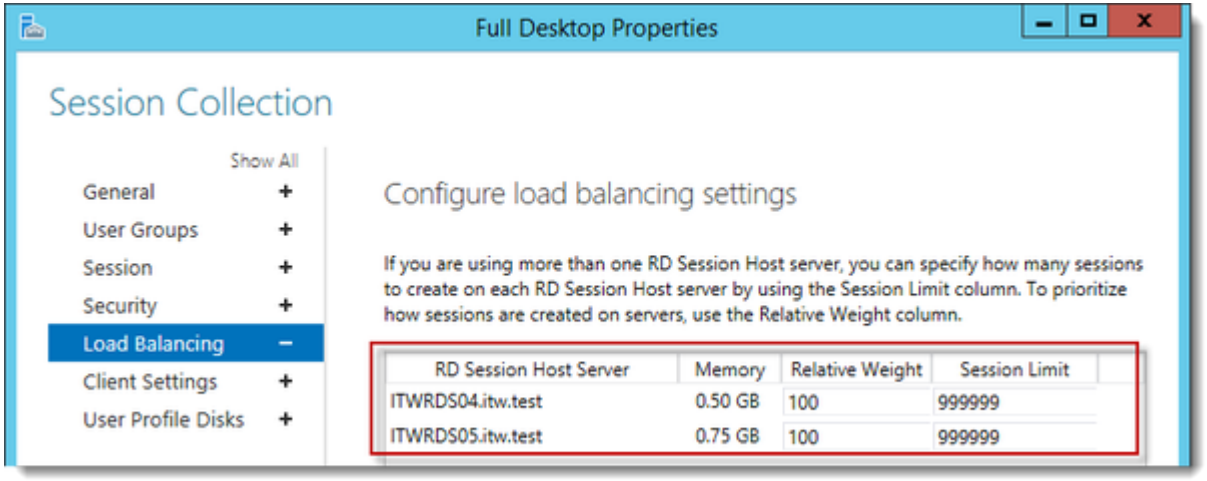
To confirm this, and see how we can influence the load balancing properties go back to Server Manager and click Remote Desktop Services, then click the Full Desktop collection.

Full Desktop collection



Click tasks, then select Edit Properties.

Session Collection



In this load balancing setup both servers are equally weighted for sessions. You could re-balance this if hardware resources are not the same across all servers in the collection.

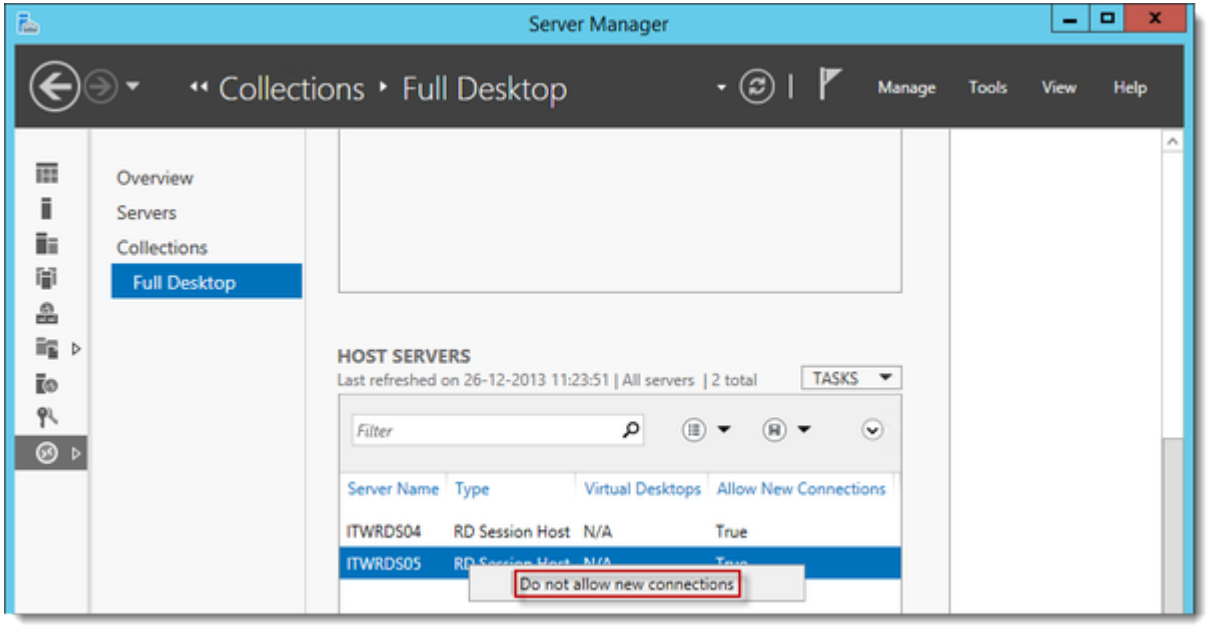
While you're in this screen, review the other properties of this session collection.

In this example we load balanced a Full Desktop session collection, but the steps to take for doing so is exactly the same for load balancing a RemoteApp program collection.

Managing a load balanced collection

Load balancing a collection makes it possible to do maintenance on your servers without annoying your users. You can put a server in maintenance without disrupting functionality.

In the Host Servers section for the collection right-click the server you want to do maintenance on.



Then select Do not allow new connections.

Of course, you will have to wait until existing sessions are completed, or instruct users to log off and log back on, in which case they will be redirected by the RD Broker to the other server. Yes, this is a new session, there is currently no way to migrate sessions to other hosts without annoying the user.

If you want to continue building along with this series, remove everything that's installed in this guide. You can revert to snapshots, or remove everything manually.

– Remove the server from the session collection.

– Remove the server from the RD deployment, removing the role services as well.

– Remove the server from the RD Gateway Resource group

– Remove the server from the domain

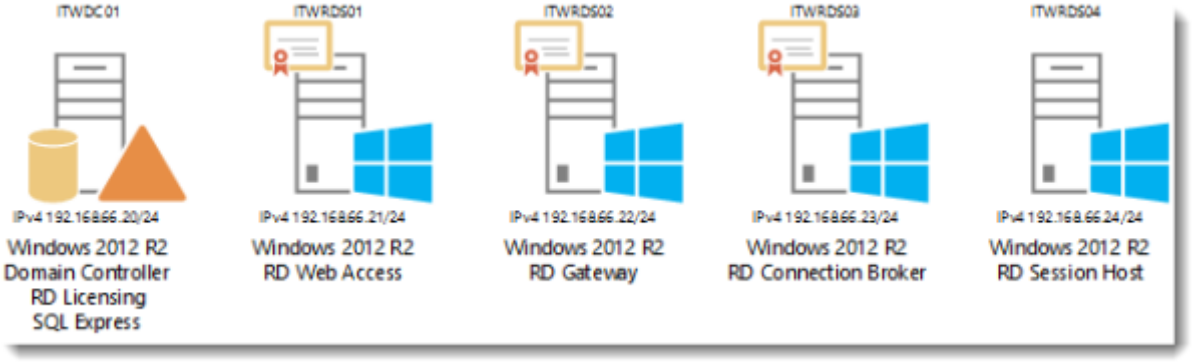
And I will see you in the next part in which I will finally show a step by step guide on deploying and publishing a RemoteApps program collection.

Step by Step Windows 2012 R2 Remote Desktop Services – Part 4

A step by step guide to build a Windows 2012 R2 Remote Desktop Services deployment.

Part 4 – Publishing RemoteApp programs

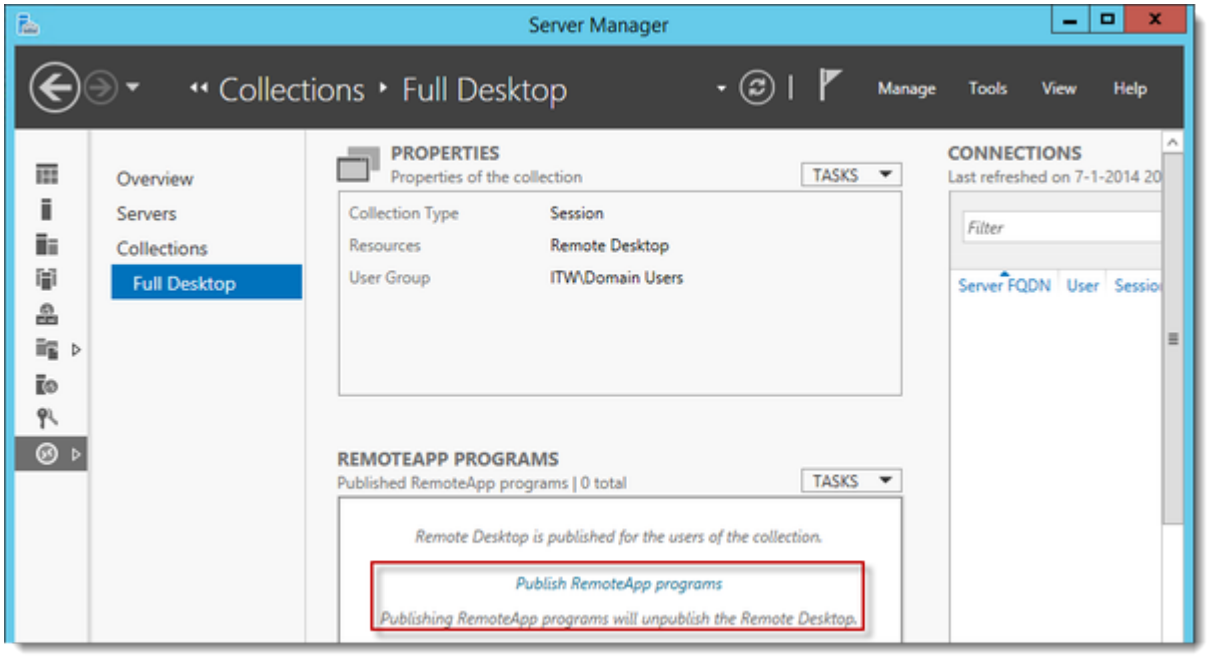
.I'll be using the setup I demonstrated in Part 2 – Deploying an advanced setup because this setup was still on my Windows 8.1 Hyper-V setup. As a reminder, here's the setup again:



Everything is up & running, so this guide won't be focusing on building the Remote Desktop Services deployment itself.

Perparing for publishing a RemoteApps collection

By the end of Step 2 in this series I had a Full Desktop session collection fully functioning. To prepare the lab for RemoteApps I can simply click the Full Desktop session collection and click the "Publish RemoteApp programs" link as shown in this screenshot:

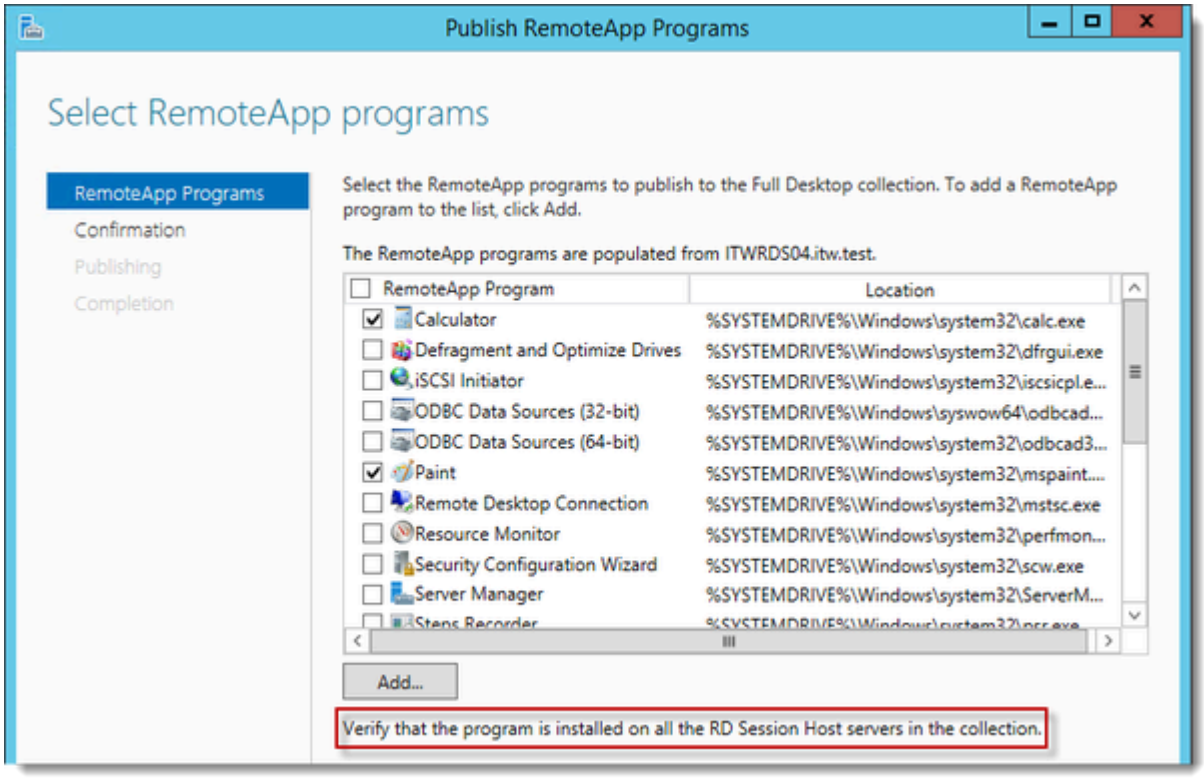


Doing so will convert the Full Desktop session collection to a RemoteApp programs collection, as mentioned in the remark below the link.

Publishing a RemoteApps collection

Click the Publish RemoteApp programs link.

Select RemoteApp programs



Immediately you are presented with a list of available applications. If you have multiple servers in the collection pay attention to the text I highlighted in the screenshot.

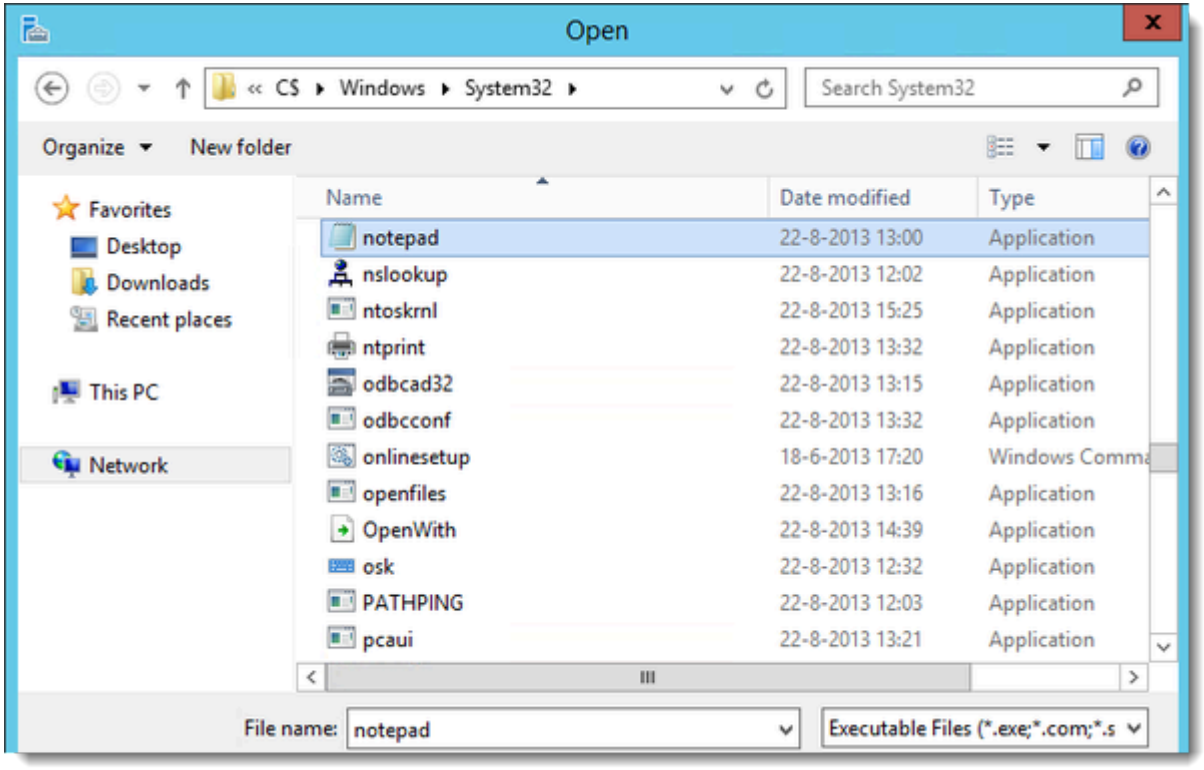
If you want to publish programs that are not in this list use the Add button to browse to the program you want to publish. Note that you need to browse to a UNC path, not a local disk on the RD Session Host.

I selected Calculator, Paint and Wordpad.

As you can see, Notepad is missing by default.

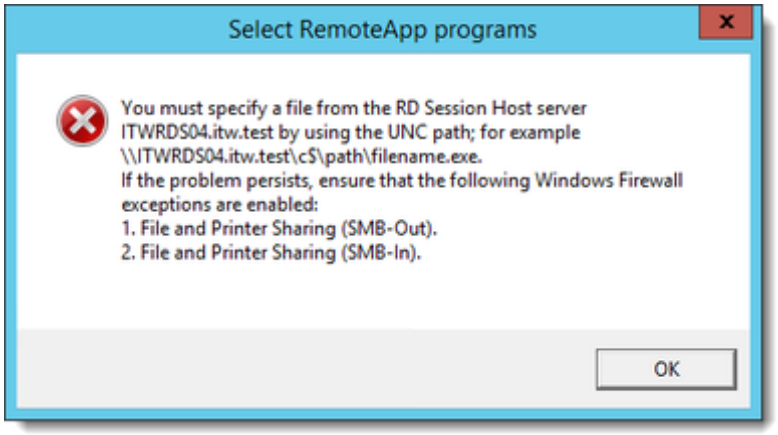
Click Add.

Open



Browse to \\itwrds04\c$\windows\system32 and select notepad.exe there.

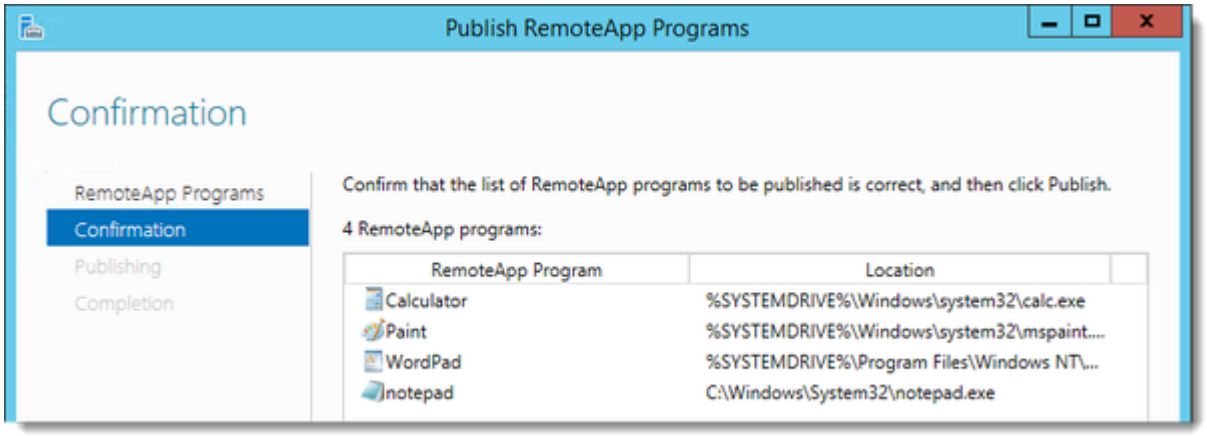
If I browse to C:\Windows\System32 and select notepad.exe:



So browse to Notepad.exe using the UNC path and click Open.

Click Next.

Confirmation

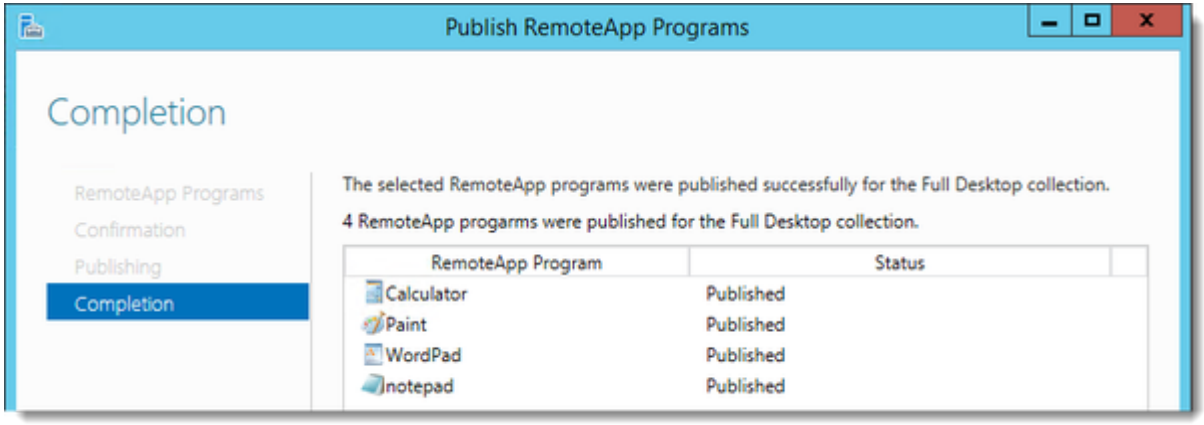


On the Confirmation page you can see the UNC path is no longer visible, but is now shown as the actual path.

Click Publish.

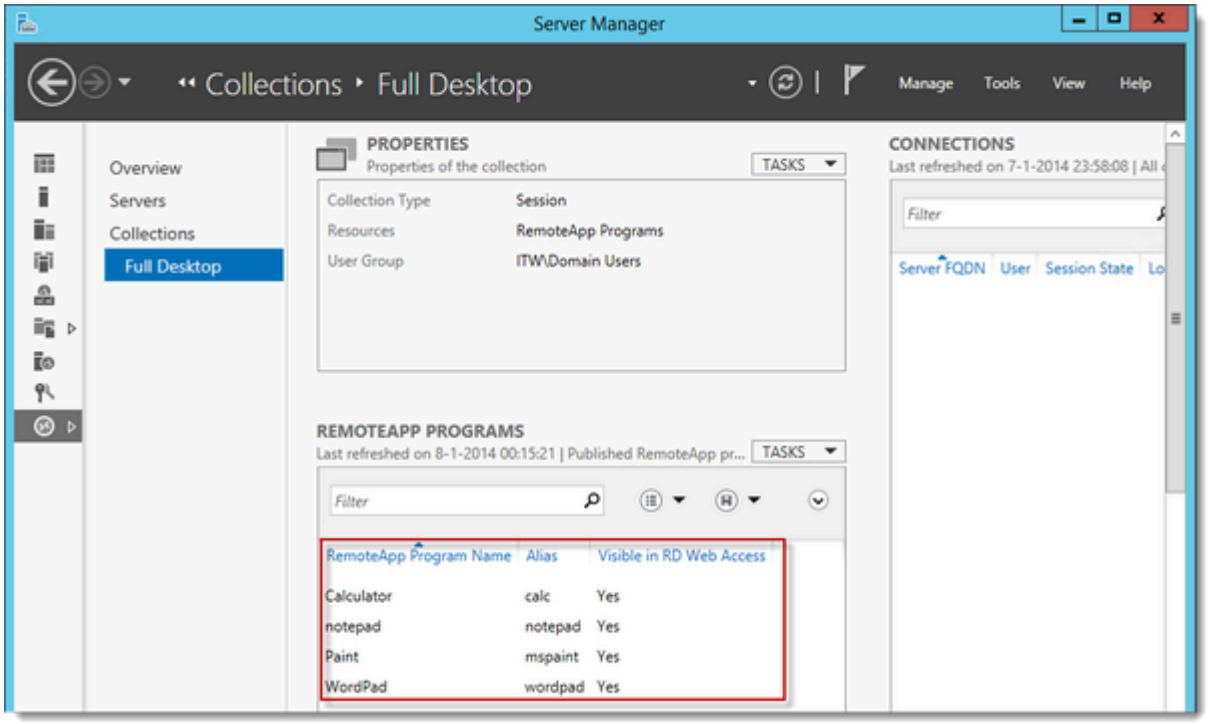
The applications you selected will be published.

Completion



Click Close.

Server Manager

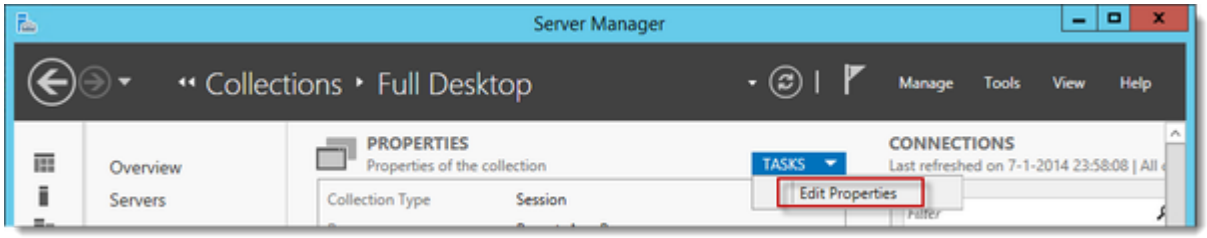


You'll return to Server Manager and you can see the applications that were just published in the RemoteApp programs sections, including basic properties like Alias and Visible in RD Web Access.

Let's finish the collection.

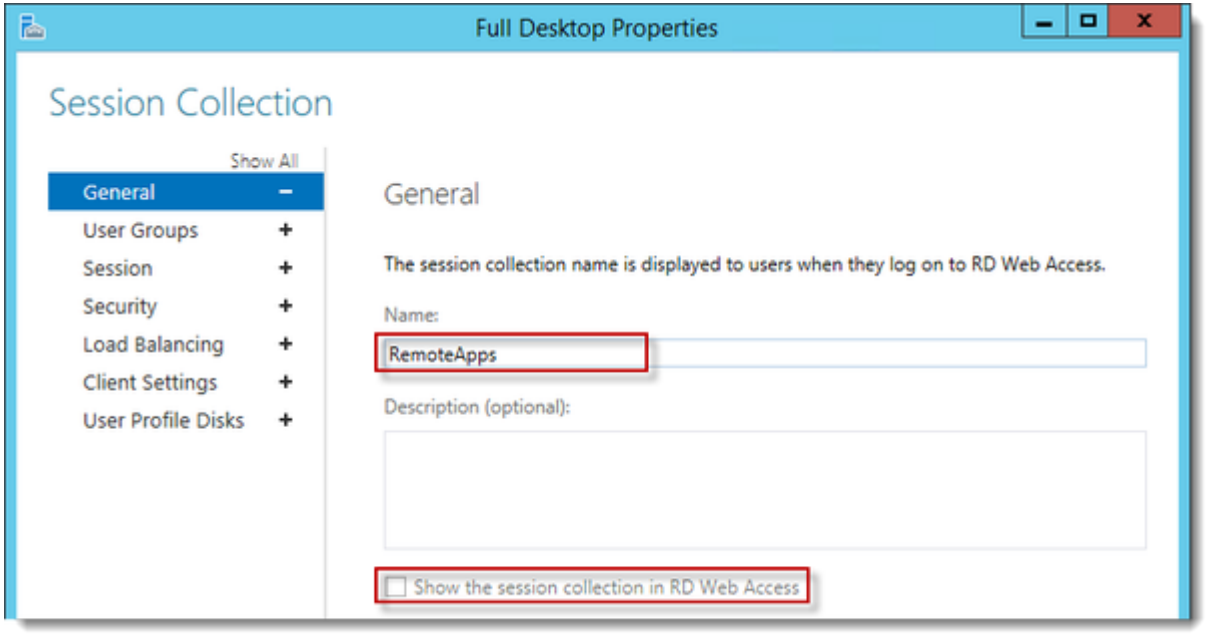
Finishing the RemoteApp programs collection

Server Manager



In the properties section for the Full Desktop collection click Tasks and then click Edit Properties.

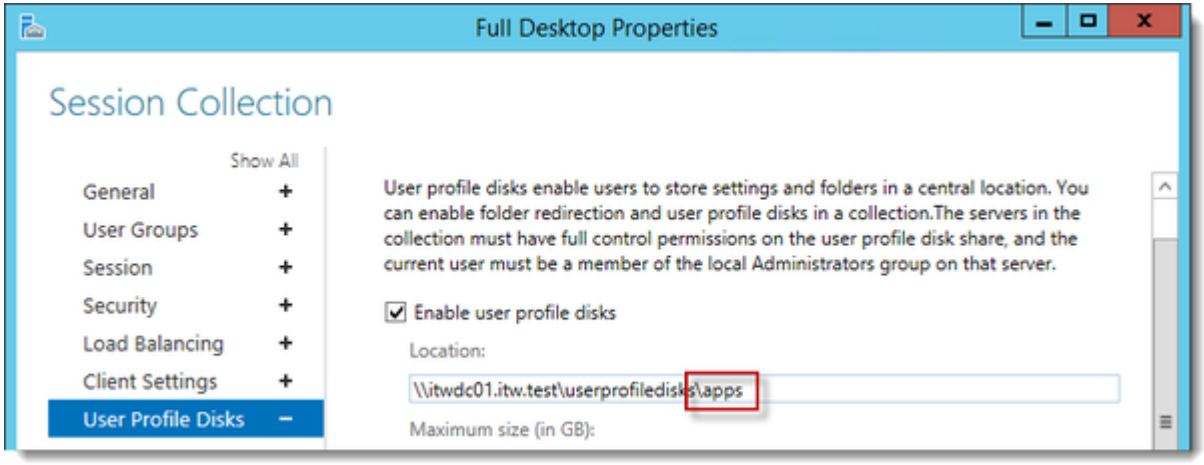
Session Collection



Rename the collection to something more meaningful than “Full Desktop”. Also notice that “Show the session collection in RD Web Access” is now greyed out since it’s no longer a session collection.

Click Next.

User Profile Disks

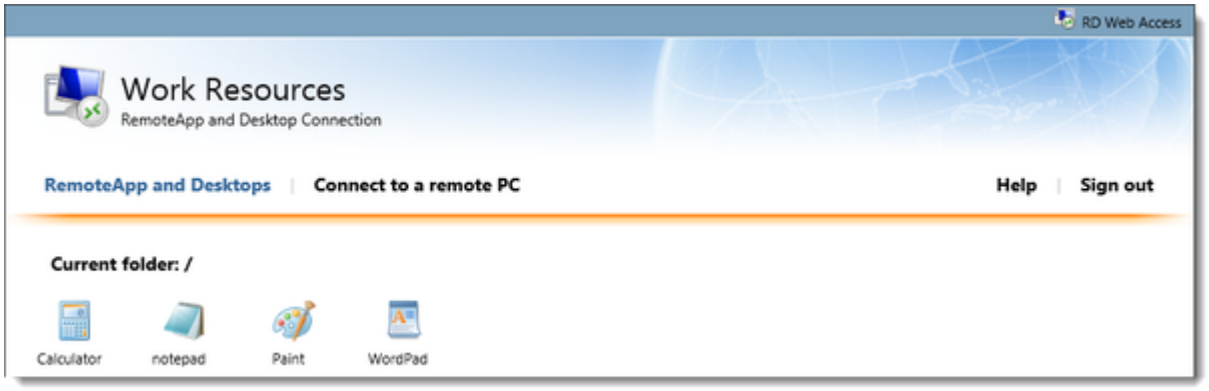


Review the settings in User Groups, Session, Security and Load Balancing, and adjust the settings in each section to your likings.

In User Profile Disks I changed the profile disks location to a different folder. Although that's not really necessary in this setup it's good practice to give each type of collection its own location for profile disks. Especially so if you're planning for multiple types of collections in a single deployment. The reason I do this is because profile disks can't be shared across types of collections. That's right. You can't. This means that if you have a deployment that supports Virtual Desktop Infrastructure (VDI), Remote Desktop session collection(s) and RemoteApp programs, you'll have three different profile disks for each user. In deployments with a large number of users you'll quickly see the need for a nice little tool like Sidder ;)

Click OK.

Now log in to the RD Web Access:

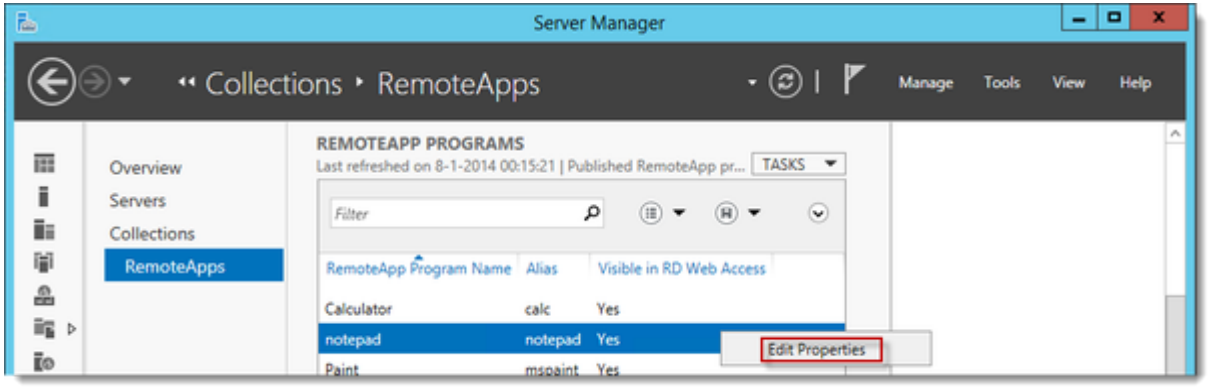


It works, but we're not done yet.

Editing a RemoteApp program

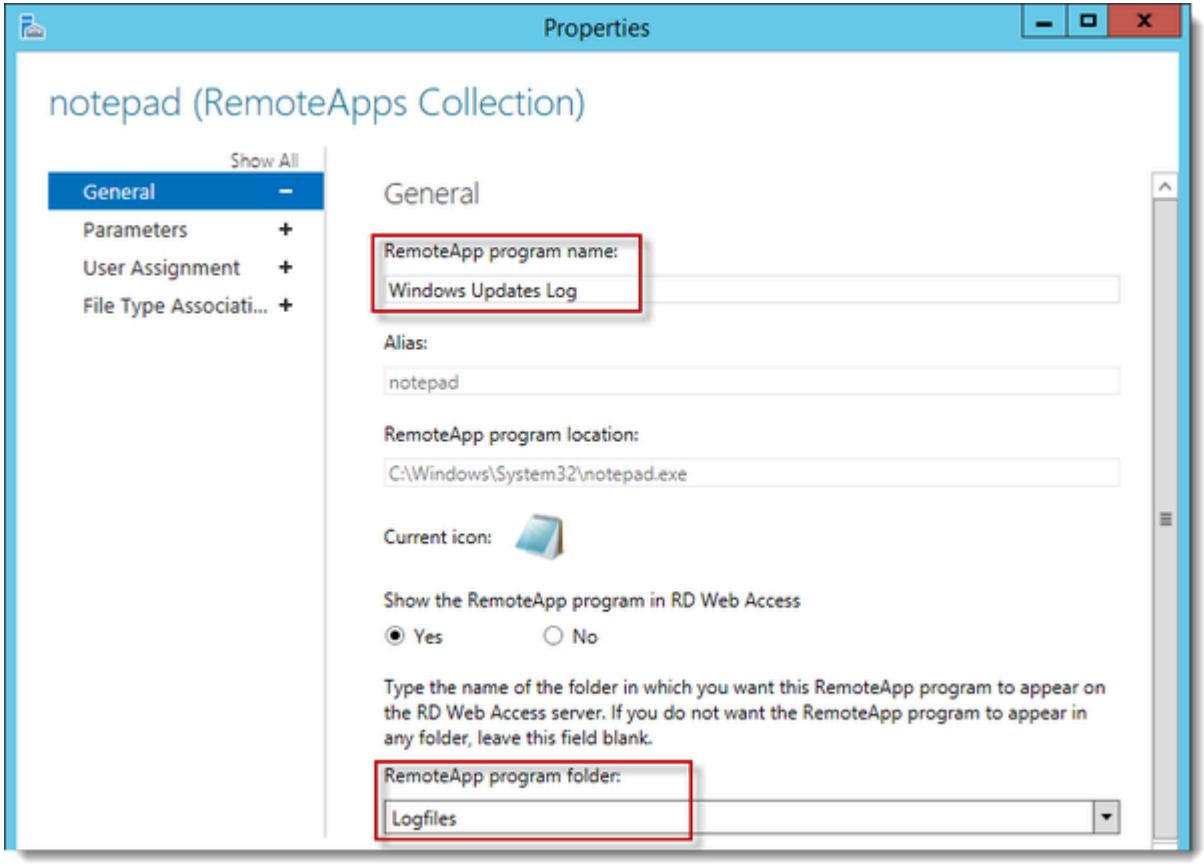
When we added Notepad.exe in the Wizard it created a RemoteApp called "notepad". Let's use this RemoteApp to demonstrate what we can manage for RemoteApps.

Server Manager



In the RemoteApp programs section, right-click notepad and click Edit Properties.

General



On the General page we can edit several attributes for our notepad RemoteApp.

We can change the RemoteApp program name. This is the name that is displayed in RD Web Access. Change this to “Windows Updates Log”.

We cannot change the RemoteApp’s alias here. You can only change the alias by deleting the RemoteApp and re-creating it using Powershell. More on that later.

We cannot change the RemoteApp's program location here.

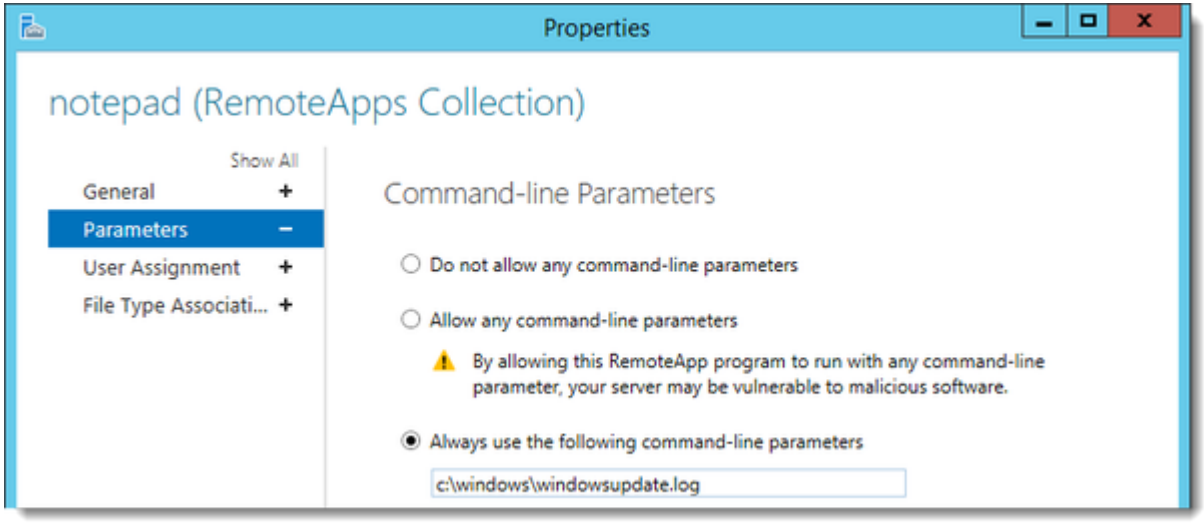
We cannot change the current icon here.

We can select to hide or show the RemoteApp in RD Web Access.

And we can select a Folder for the RemoteApp. If you click the dropdown menu you'll notice it is empty. Don't worry, just type in the folder name. Enter "Logfiles" here. This is the way to add new folders. If you have created folders before, you can select them using the dropdown menu.

Click Parameters.

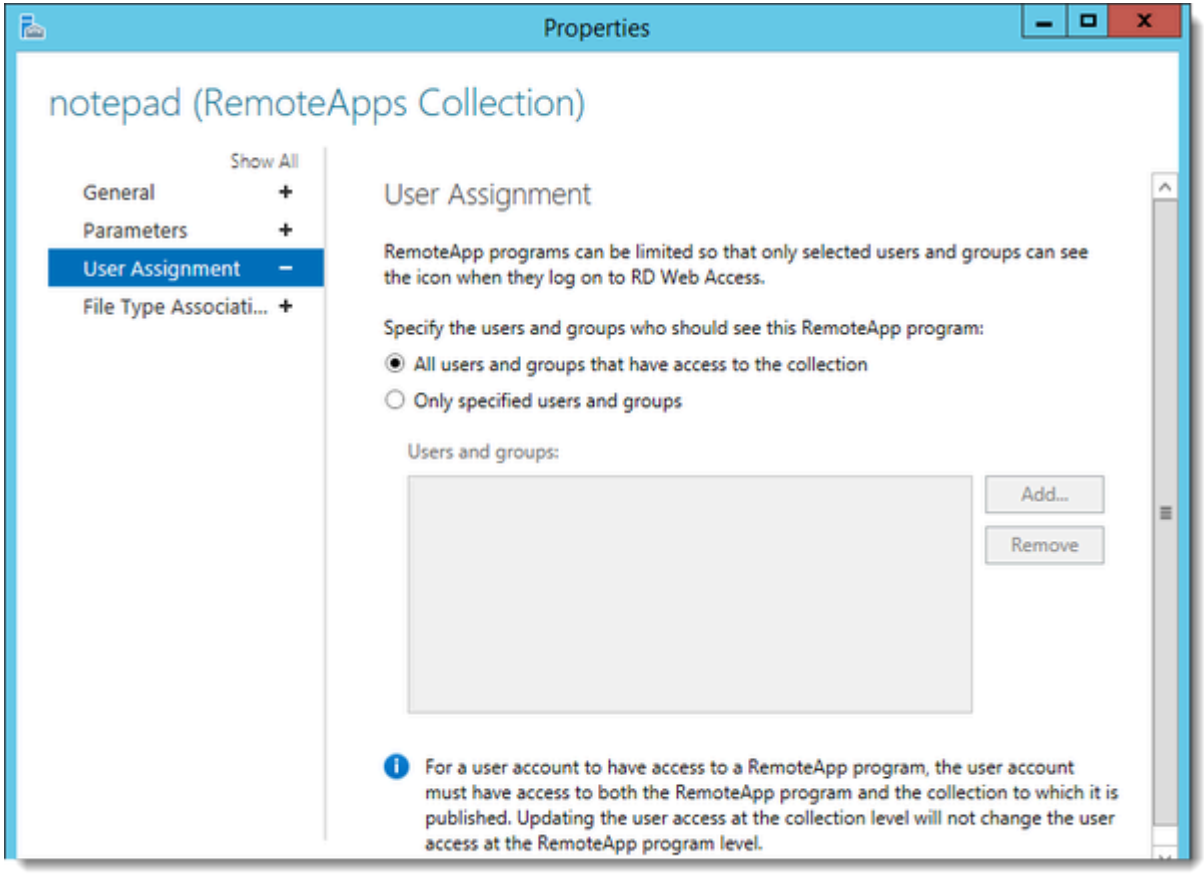
Parameters



If your RemoteApp program needs any parameters to run, this is the place to enter them. Enter "c:\windows\windowsupdatelog" for this one.

Click User Assignment.

User Assignment

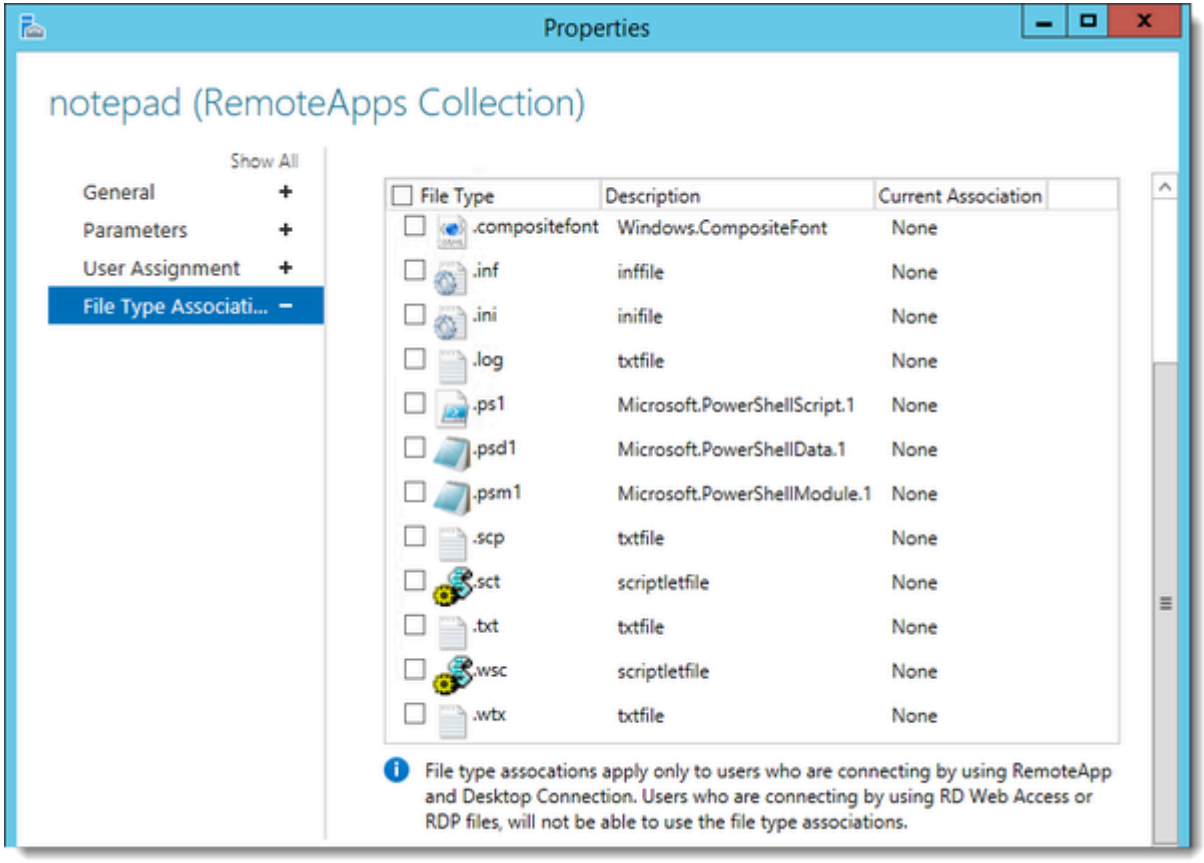


You can fine-grain user assignment on RemoteApp program level. For example, you can publish the complete collection to Domain Users, but limit this application to Domain Admins or Log Admins. In this case the Logfiles RemoteApp folder will be hidden for Domain Users as well, since this is the only application in this folder.

Review the remark in the bottom of the screenshot.

Click File Type Associations.

File Type Associations

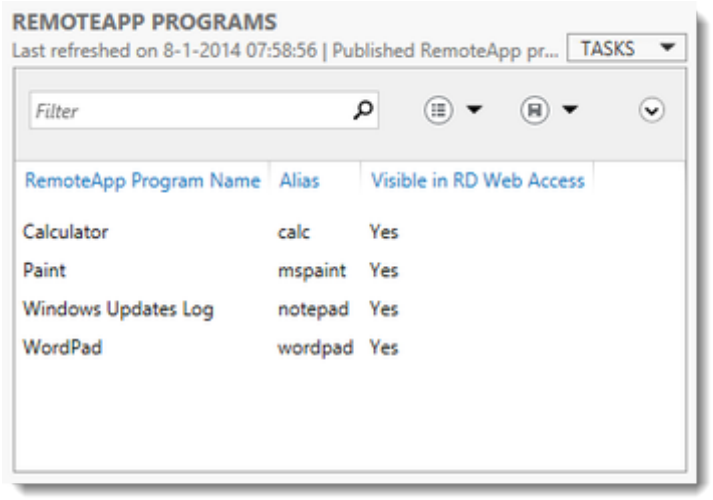


You can set desired File Type Associations for your RemoteApp program here. Take notice of the remark when you scroll down. What this means is that associations will only take effect if the RemoteApp is started through “Connected RemoteApp and Desktop Connections”, and not if you start it using custom RDP files, or through RD Web Access.

Since we just changed this one to publish the Windows Update Log, we don’t need any File Type Associations, I’ll get back to this later.

Click OK.

Review the list of RemoteApp programs and notice the change in RemoteApp Program Name:



Refresh or log in to the RD Web Access and review these changes:

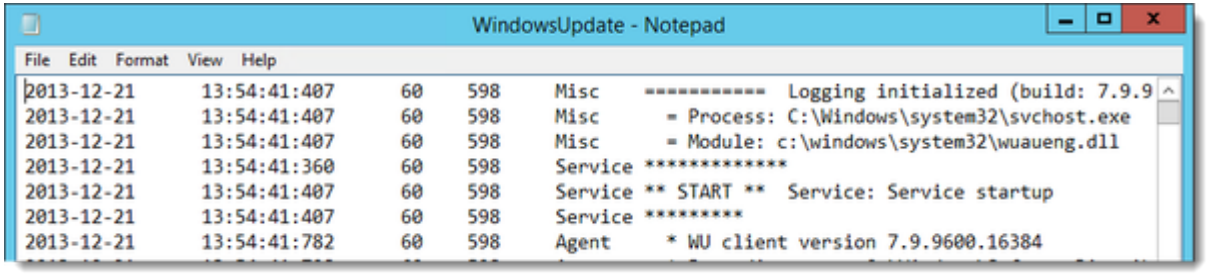
Here’s the folder we entered.

Clicking the folder shows the RemoteApp we just published.

Note: if you want to customize views like these, check out another step by step series I am publishing.

Click Windows Updates Log.

Windows Updates Log



And it works.

Using Powershell to manage RemoteApp programs

Get-RDRemoteApp (http://technet.microsoft.com/en-us/library/jj215454.aspx) is used to list properties for RemoteApps.

Example:

```
Get-RDRemoteApp -alias "wordpad" | fl
```

Set-RDRemoteApp (http://technet.microsoft.com/en-us/library/jj215494.aspx) is used to set properties for RemoteApps.

Example:

```
Set-RDRemoteApp -Alias "wordpad" -DisplayName "WordPad - Renamed"
```

New-RDRemoteApp (http://technet.microsoft.com/en-us/library/jj215450.aspx) is used to create a new RemoteApp in a certain collection.

Example:

```
New-RDRemoteApp -CollectionName "RemoteApps" -Alias "regedit" -DisplayName "RegEdit" -FolderName "Admin Tools" -FilePath "C:\Windows\regedit.exe"
```

Remove-RDRemoteApp (http://technet.microsoft.com/en-us/library/jj215493.aspx) is used to remove a RemoteApp.

Example:

```
Set-RDRemoteApp -CollectionName "RemoteApps" -Alias "wordpad"
```

Get-RDAvailableApp (http://technet.microsoft.com/en-us/library/jj215457.aspx) is used to list available applications to publish in a collection.

Example:

```
Get-RDAvailableApp -CollectionName "RemoteApps"
```

Get-RDFileTypeAssociation (http://technet.microsoft.com/en-us/library/jj215461.aspx) lists the filetype association(s) for a certain application.

Example:

```
Get-RDFileTypeAssociation -AppAlias "wordpad"
```

Set-RDFileTypeAssociation (http://technet.microsoft.com/en-us/library/jj215459.aspx) is used to set the filetype association(s) for a certain application.

Example:

```
Set-RDFileTypeAssociation -CollectionName "RemoteApps" -AppAlias "wordpad" -FileExtension ".txt" -IsPublished $True
-IconPath "%ProgramFiles%\Windows NT\Accessories\wordpad.exe" -IconIndex 0
```

And that concludes this step by step on publishing RemoteApp programs.

In the next part of this series I will show how to use and configure the “Connected RemoteApp and Desktop Connections” in combination with this setup.

https://msfreaks.wordpress.com/2013/12/09/windows-2012-r2-remote-desktop-services-part-1/