

Installing a new Windows 2012 R2 Remote Desktop Services (RDS) Server

Server installation

I've been thinking about renewing our old Terminal Server 2003 for a loooooong time (since the [migration of VMware to Hyper-V](#)). We had licensing problems with Windows Server 2008, so I thought I'd give it a new try with Windows Server 2012 R2. Nothing much to it really, here are the steps:

- Installed our basic image of Windows Server 2012 R2 Standard
- Applied local security policies (exported from Security Compilcane Manager 3.0) with LocalGPO tool (optional extra protection apart from our GPOs).
 - The tool wasn't updated for Windows Server 2012 R2, only for Windows Server 2012. The script checks for OS version so it can't continue if wrong OS is detected. Well, theoretically at least. I edited the script and commented out the line which checks for OS version (in LocalGPO.wsf).
 - I replaced "Call ChkOSVersion" with " ' Call ChkOSVersion" (that is, commented out the whole check).
 - Ran the script:
 - C:\Program Files (x86)\LocalGPO>cscript LocalGPO.wsf /path:c:\Users\xxx\Downloads\gpo\{3efc9336-94bb-4719-b14d-c34b8121f86f}
Microsoft (R) Windows Script Host Version 5.8
Copyright (C) Microsoft Corporation. All rights reserved.

Modifying Local Policy... this process can take a few moments.

Applied valid INF from c:\Users\xxx\Downloads\gpo\{3efc9336-94bb-4719-b14d-c34b8121f86f}

Local Policy Modified!

Please restart the computer to refresh the Local Policy

- Worked!
- Installed the RDS Server role (Fig 1) following a guide from <http://www.techieshelp.com/windows-server-2012-install-and-configure-remote-desktop-services/> (my own knowledge was a bit rusty)
- I used Quick Start and Session-based Desktop Deployment (no need for "full" virtual machine-based deployment in our case)
- Remote Desktop Services RemoteApp (RD Web Access) also got installed, but I won't use that feature for now.

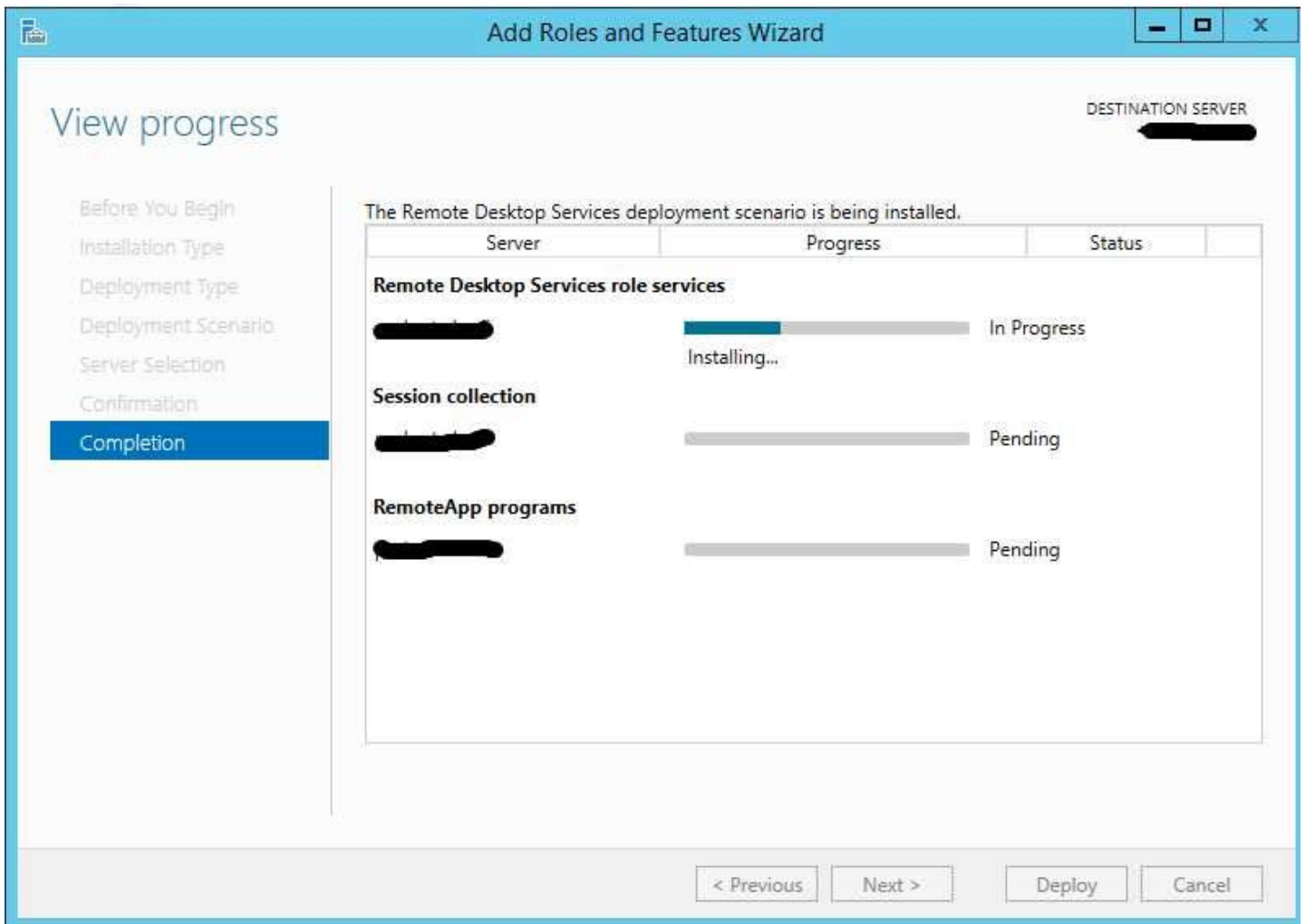


Fig 1. Installing the roles... yawn.

- Next step is licensing. A RDS Server requires licenses either per device or per user (Remote Desktop CALs). Our University has some licenses so no problem. I won't go into the licensing system here, I just say that we'll be using per machine licenses and that the licenses will be issued from an existing license server in our domain. The procedure for adding the license server was a bit different from the one in the above guide though. In the guide it says "Click the RD licensing icon and either add the server as your license server *or point it to your existing license server on the network* by entering the server name or IP then click the forward arrow". I couldn't point to an existing license server in this step. Oh well, the steps are still quite similar, but I skipped to the next step in the guide. My steps in Fig 2.

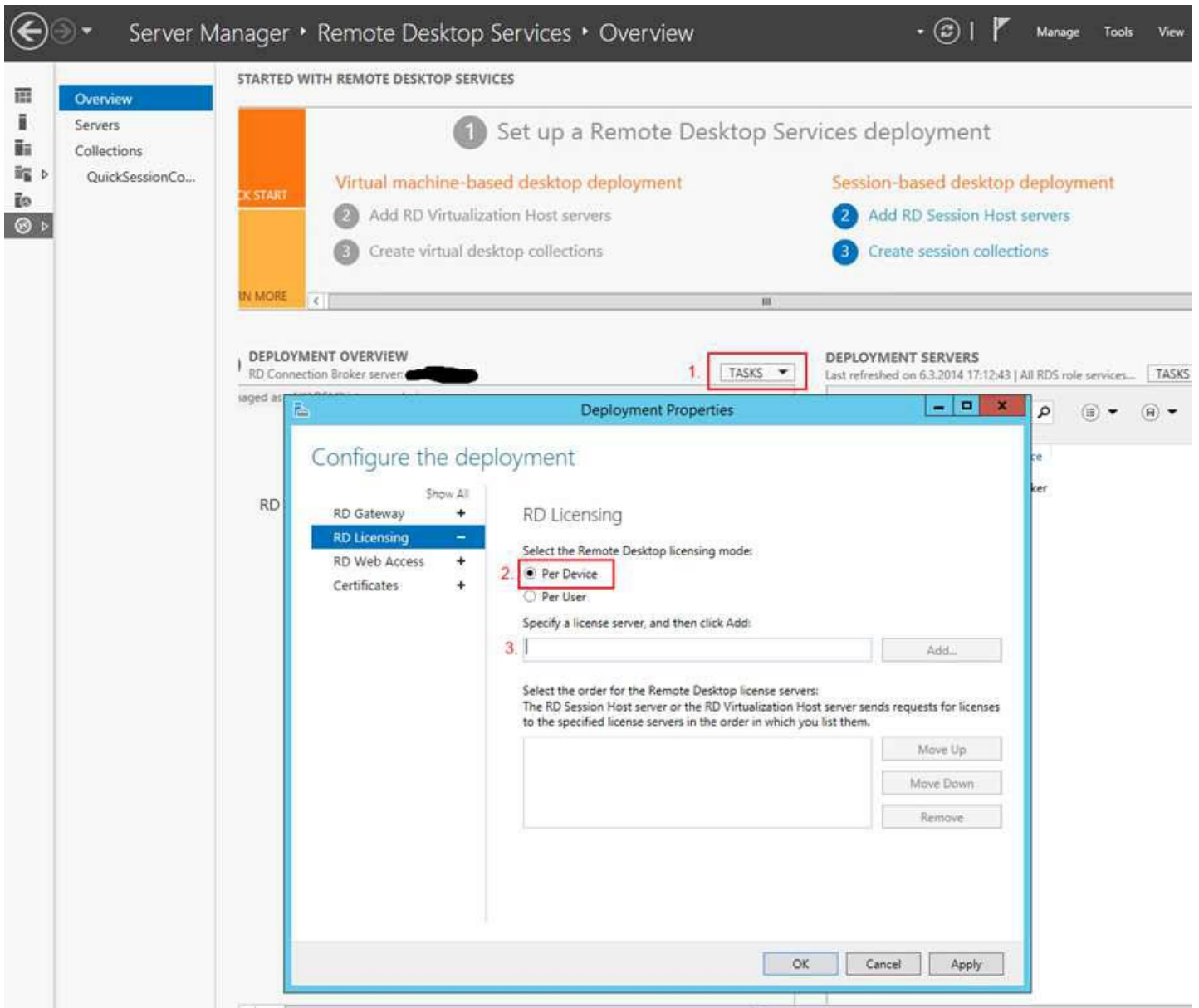


Fig 2. RDS licensing.

1. TASKS, Edit Deployment Properties
2. Per Device
3. Enter license server
4. Done!

If you are interested in the difference between per user and per device licenses, read <http://technet.microsoft.com/en-us/library/cc753650.aspx> for example.

Checking that the licensing is working can be done from RD Licensing Diagnoser (I didn't install RD Licensing Manager, it is already installed on another (license) server). Everything seemed ok! (Fig 3 has the details).

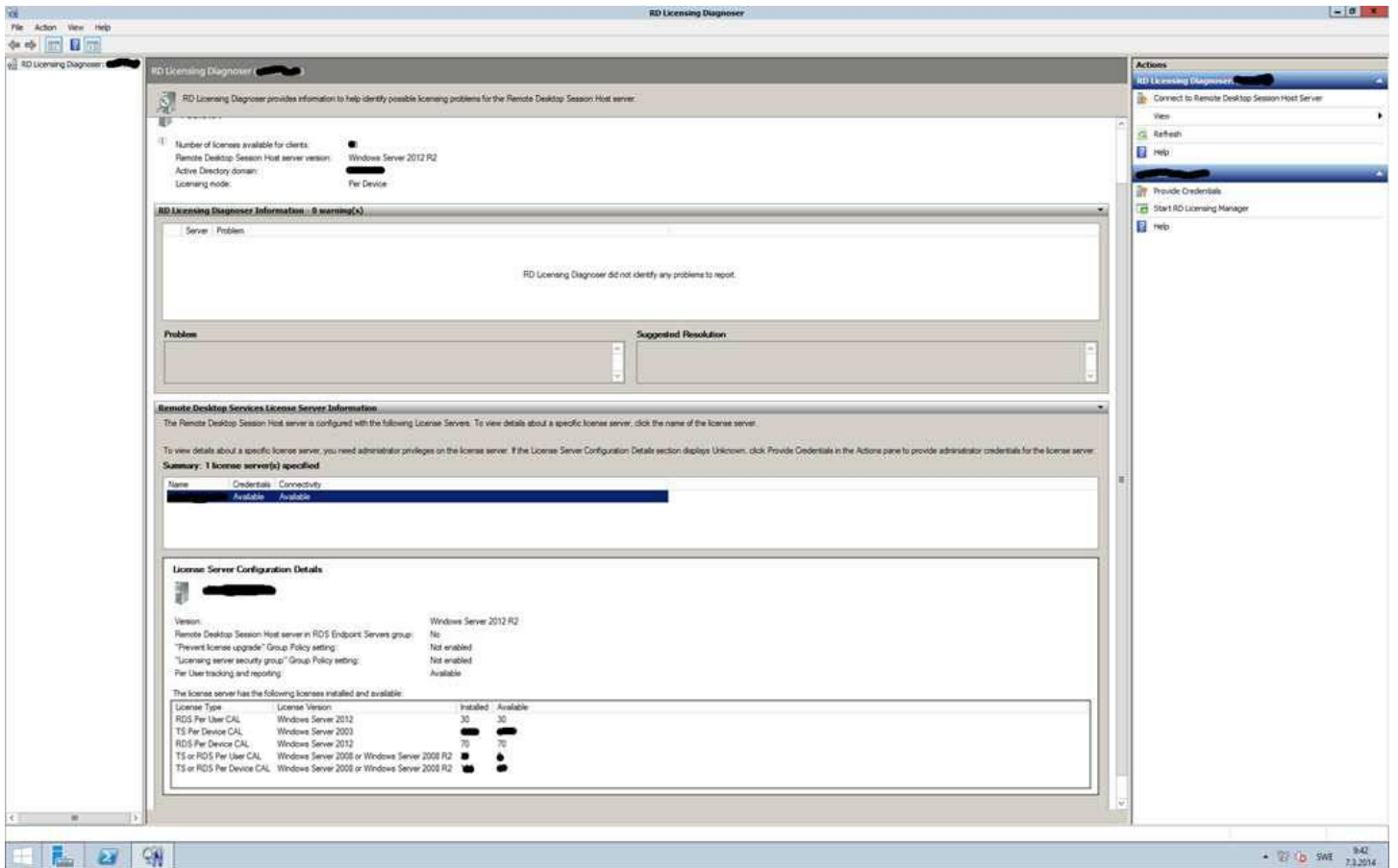


Fig 3. RD Licensing Diagnoser.

Checking that a client gets a (permanent) license can be done from the license server/RD Licensing Manager (Fig 4).

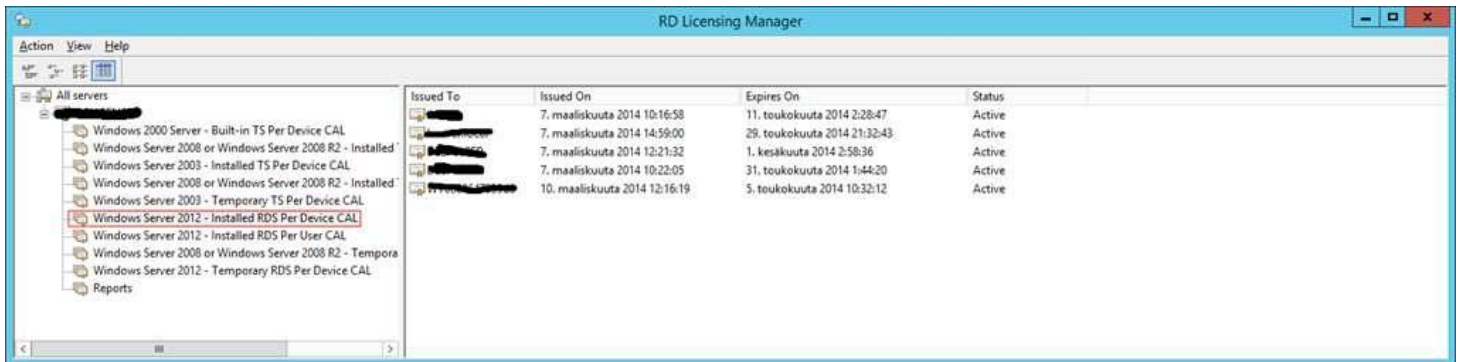


Fig 4. RD Licensing Manager. Five (per device) licenses are currently checked out

Client configuration

Now that the licensing part seems to be working just fine, it's time to test out different clients. We are using Windows, Linux and Mac clients. We also have a small classroom with WySe Thin Clients. For once, there were no problems

connecting from Linux clients Mac on the other hand was playing hard, but the best (and easiest solution) is to use the newest Remote Desktop Client from Microsoft (available in the app store, <https://itunes.apple.com/en/app/microsoft-remote-desktop/id715768417?mt=12>). This works out of the box. If you are interested in the problems with the Remote Desktop Client bundled with MS Office 2011 for Mac, read:

<http://blog.mikejmcguire.com/2013/10/15/r2-d2-you-know-better-than-to-trust-a-strange-computer-why-doesnt-the-mac-os-x-rdp-client-trust-windows-server-2012-r2-2/>

I've noticed that there can be problems with Network Level Authentication and RDP however. Even though it's probably not the best idea to disable it, I've done it. It saves you a LOT of headache. Older clients are able to connect without problems when this is disabled.

The WySe Thin Clients had an old firmware which didn't support the RDP security of the new Windows Server 2012 R2. I've upgraded the firmware in them before, so I knew that this was probably the solution (and it was). The steps for upgrading the firmware (short version):

- Download newest firmware (it's not free, you have to have an agreement with a vendor)
- Upload the firmware file (VL10_wnos in our case) to the /var/ftp/wyse/wnos –directory on the ftp server (or similar path on a Windows Server)
- Change the wnos.ini file to upgrade its firmware on reboot, "Autoload=2" (# Selects firmware update mode: 0=disable,1=upgrade/downgrade, 2=upgrade only)
- After upgrading the firmware, put the flag back to "0" (disable).
- Done. Windows Server 2012 R2 compatible WySe Thin Clients

Server configuration

There's nothing much to do really if the "underlying work" is done. With this I mean using existing group policies and so on. We already have a group policy that enables folder redirection so the users home directories won't get stored locally (only cached locally). This way you can easily upgrade/reinstall the server without losing user data. I'm not going to write about different GPOs and folder redirections, because it's outside the scope of this post.

The server was "empty" so I had to Install applications. You shouldn't install applications "the regular way" on a RDS Server, instead you should use the special install mode called **RD-Install**. You can do the mode-switch either by command prompt or graphically. Details here: <http://technet.microsoft.com/en-us/magazine/ff432698.aspx>. Nothing

much to tell about the process of installing applications, about as exciting as watching paint dry Just install according to your users needs.

I also installed **Classic Shell** (yes, I'm a fan) as I don't want the new users to get annoyed and confused about the new (crappy) metro interface. I was able to force the same settings for every user with policies. There are (group) policy definitions available in C:\Program Files\Classic Shell\PolicyDefinitions.zip (from version 4.0.4 onward). Unzip them to C:\Windows\PolicyDefinitions (or put them on your domain controller) and play around until you find the settings for your own needs.

Settings that I changed for Classic Shell:

- Show Metro Apps: enabled (no mark in the Show Metro Apps checkbox, which means that the Metro apps aren't shown).
- Disable active corners: enabled (Disable active corners: All)
- Enable accessibility: enabled (no mark in Enable accessibility checkbox)
- Enable settings: enabled (no mark in checkbox, which means users can't change Classic Shell settings)
- Menu items for the Windows 7 style: enabled (I modified the menu to my preference, and then copied the the registry settings from HKCU\Software\IvoSoft\ClassicStartMenu\Settings and copy/pasted the settings here.
- Shift+Win opens: enabled (Shift+Win opens Windows Start menu, that is, metro)
- Button look: enabled (Button look: Aero button)
- Show Start screen shortcut: enabled (no mark in the checkbox)
- Windows key opens: enabled (Windows key opens: Classic Start Menu)

If you want to skip the metro screen by default (I surely do), use group policy preferences:

<http://www.petri.co.il/bypass-start-screen-windows-8-1-server-2012-r2.htm>. This setting is also available in Classic Shell, but it doesn't work in a RDS environment without using group policy preferences.

Furthermore I disabled write access to the root of the C:\ drive. Users shouldn't be able to write stuff on the RDS Server, only in their own profiles. You can do this by going to the root of the C:\ drive and select Properties → Security. Then under Security choose Advanced. Remove all the rights from "Users" except Read & Execute. See Fig. 5.

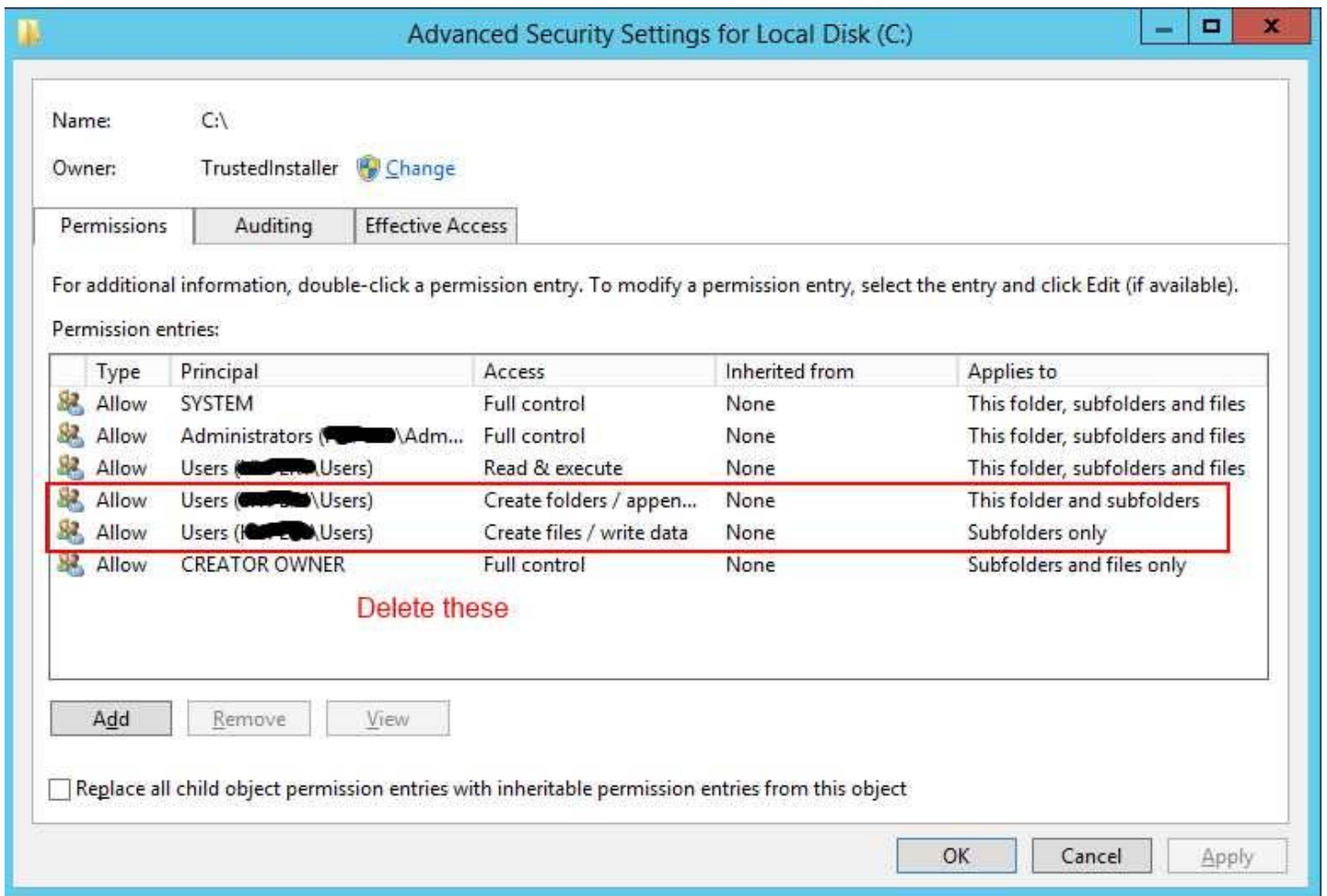


Fig 5. Advanced Security Settings

You will receive many errors about setting file permissions (Fig 6 for example), which is normal. Anyways, in the end this approach does work, and users aren't able to write to the root of the c:\ drive. Everything else also works as normal (tested before).

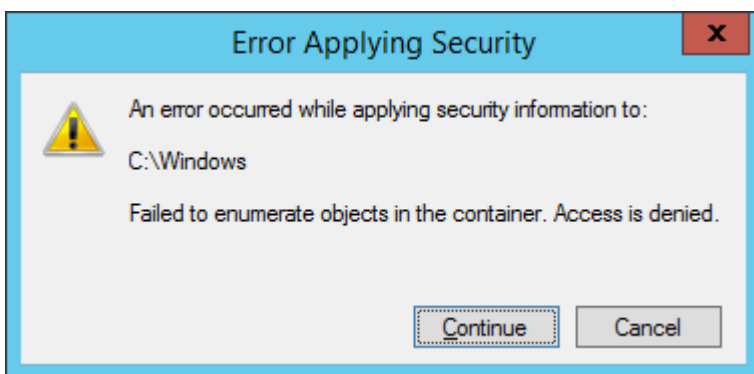


Fig 6. Error. Just ignore.

As the old server didn't have folder redirection, the users files were stored locally on the server. Luckily people didn't have that much important stuff stored on the server however, so I'll just keep a backup of the files and copy them to any user that needs their old files. As I already said, the new server is using folder redirection so nothing gets stored locally (and I won't have this problem again).

Gotchas

After the server was in production state I noticed a weird message in Server Manager → Remote Desktop Services → Overview. See Fig 7.

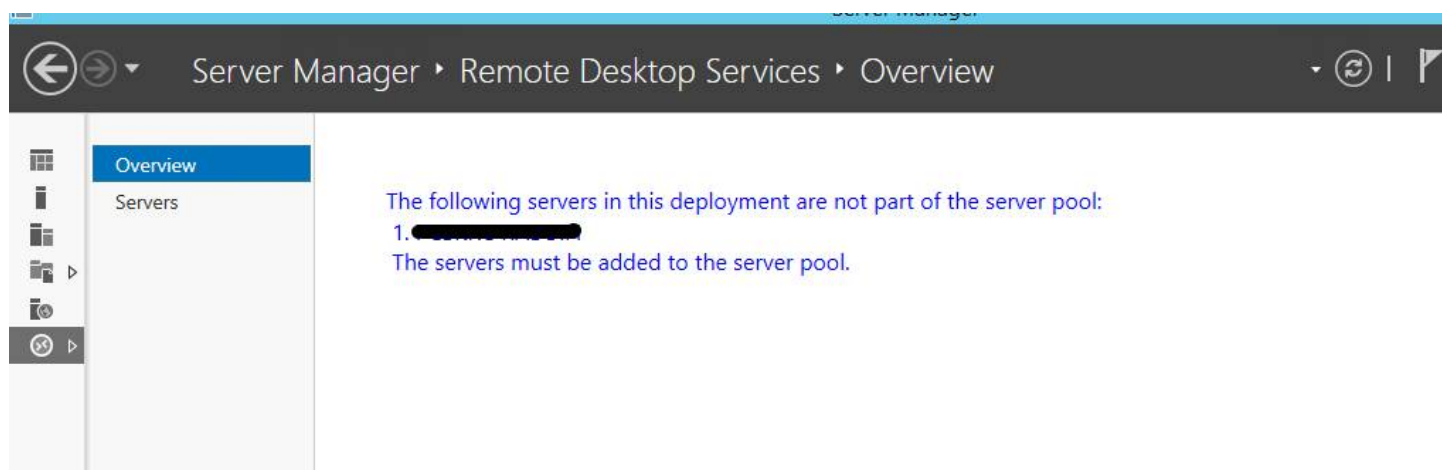


Fig 7. The following servers in this deployment are not part of the server pool:

1. [server_name]

The servers must be added to the server pool.

I knew what this was about, because I had changed the hostname of the server after testing it. Even though the server was working as it should, it seems that RDS doesn't like if the hostname/machine name gets changed after the RDS server role is installed. I did a bit of research and the solution is here: <http://support.microsoft.com/kb/2910155>. Well almost anyway. Trying the Remove-RDServer command gave me "Remove-RDServer : The RD Session Host servers are part of a session collection". Well, I didn't know much about this so I thought I'd try removing the collection. "Get-RDSessionCollection" gave me:

```
PS C:\Windows\system32> Get-RDSessionCollection
```

CollectionName	Size	ResourceType	CollectionDescription
QuickSessionCollection	1	RemoteApp programs	

I tried removing it with "Remove-RDSessionCollection". Well, good idea, but it didn't work. Actually nothing I tried worked until I saw that the server manager displayed the command "Connecting to RD-Connection Broker" and after that nothing happened. I figured the Connection Broker-part is broken. So I uninstalled the connection broker with the command "Remove-WindowsFeature -Name RDS-Connection-Broker" and rebooted. I then reinstalled the connection broker with "Add-WindowsFeature -Name RDS-Connection-Broker".

Now the Overview window in server manager told me that I didn't have any Deployments ready, and that I should add one from Server Manager Add roles and features. Well, I did the exact same steps as before (Server installation – Quick

Start and so on), and now everything was back to normal. Phew! Don't mess with the computer name.

Gotcha sources:

<http://www.virtualizationadmin.com/articles-tutorials/vdi-articles/general/using-powershell-control-rds-windows-server-2012.html>

<http://blogs.technet.com/b/manojnair/archive/2011/12/02/rds-powershell-tfm-part-iii-configuring-remote-desktop-connection-broker-using-powershell.aspx>

A final screenshot when “everything is back to normal”:

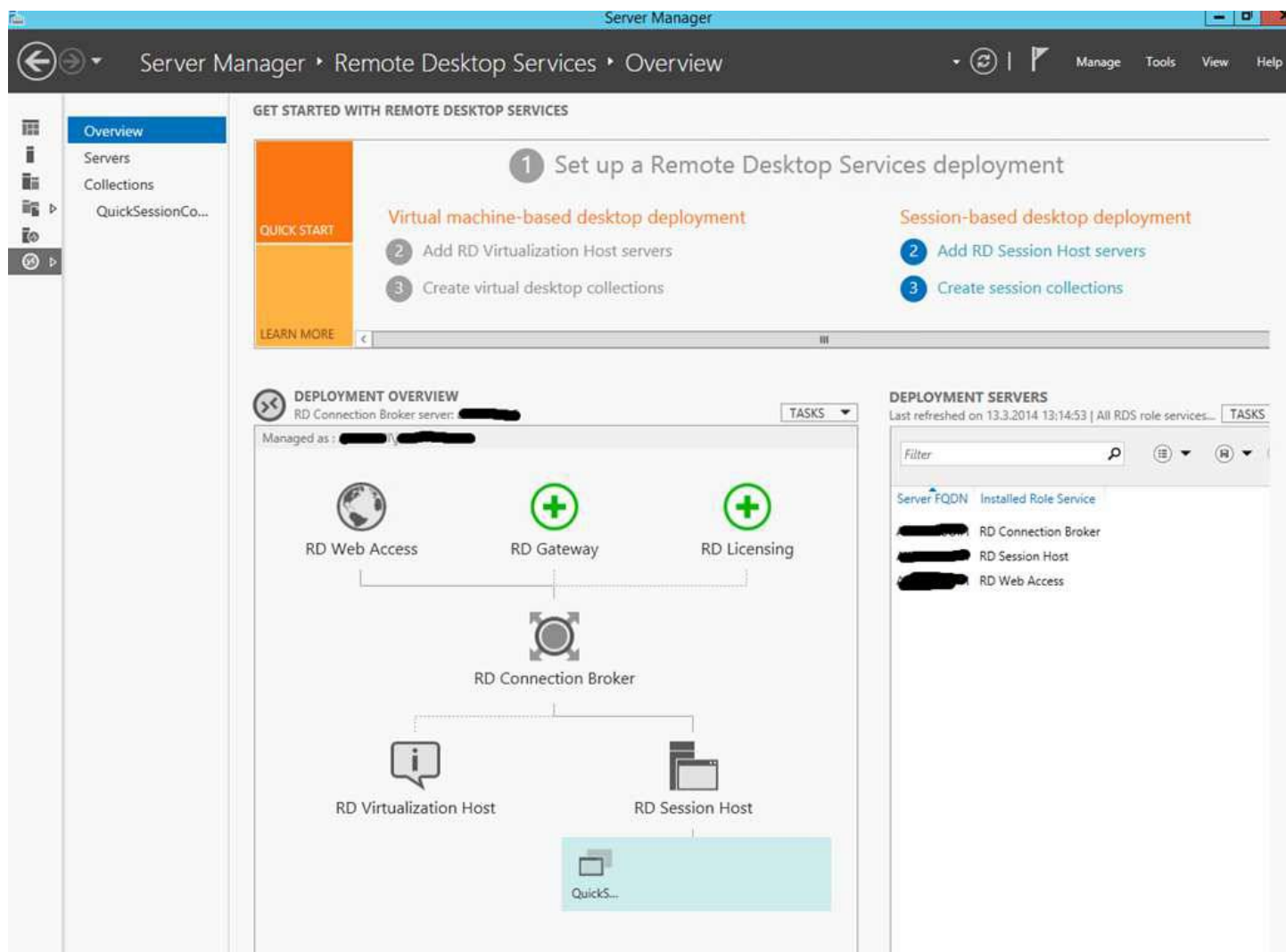


Fig 8. RDS Overview.

And another one from the Desktop:

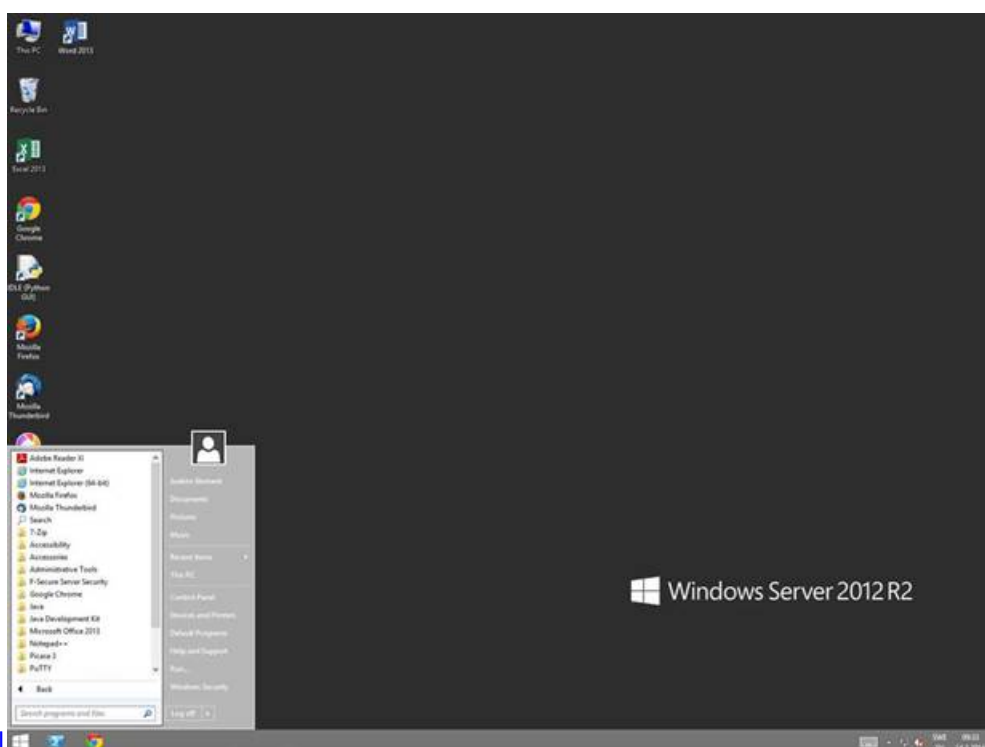


Fig 9. RDS in action

To sum it up: My deployment is Session-based desktop deployment (quick start) with licensing per device. The licensing server is on another host. RDS Web Access is disabled (even though installed) because I won't use it at the moment. I also use Classic Shell because I like it