Remote Access DB

One Developer’s Journey Creating
a Remote Access DB Solution

"Our need will be the real creator"

- Plato

My Current Access App

Access App Details:

- Name : UDS Analysis Database (UAD)
- Structure : Access 2016, Split DB (compile accdb developer front end to accde front end, accdb back end), 32 Office bit development
- Clients : 39 client purchases, 31 clients still using
- Use : Data importing (XML file) and reporting app, reports are graphical, data exporting capable (Excel), reports in pdf format
- Updates : Remote updating of app and back end tables based upon version loaded onto client, new versions stored on remote hosted server and applied by client when available
- Development : “One version” focus to avoid additional development, errors are stored in app table, reported from app via ftp, and sent to remote hosted server.
- Client Environment : You name it – they got it!

Challenges

Clients running Access app with hodge-podge of environments – Support nightmare!

- Access (Office) Version (365, 2016, 2019, 2021, ?)
- Office installed bitness (32 vs. 64)
- Windows, Office version applied (dependant on channel)
- Interaction with other program installed versions (Adobe)
- Firewall programs, antivirus software
- Installation of new app version requiring IT/Admin rights
- The need to involve outside IT resources (time coordination)
- Mirad of network configurations (remote, vm, long hops, slow speeds)
- Wondering were the backend db is?

My need

How would the world look if my Access app ran in the best possible environment:

- Reduced support time
- Allow for focused development (code only one App version for one Access version)
- Continue use of current technology used in the app (Excel, Adobe, XML)
- Have option for new technology and Access versions (include new Access graphs)
- Provide a better customer experience

But….How do I do this for many different clients who have my application running on so many different environments?

Ultimately, I found that my need was to be able to restrict a user to a common application (one version) on a consistent Windows environment.

RemoteApps Decision

After a lot of searching, looking at other development environments, and other ways to deliver the Access apps, I zeroed in on <u>RemoteApps</u>.

WHY?

- It met my main goal in that it would allow me to restrict a remote desktop user to a single application in a “controlled” environment
- I could use my existing Access application with minimal revisions to code
- The knowledge and cost barriers to development and implementation of the solution seemed low (Windows desktop app)
- Familiarity, as I had already seen an FMS presentation by Luke Chung on RemoteApps

Reference Articles:
- http://fmsinc.com/MicrosoftAccess/consulting-services/remoteapp-remotedesktop.html

Solution – Remote Apps?

Hasn’t that solution gone away?

No. RDS/RemoteApp is still being developed and supported. It runs as a solution on Azure IaaS

Microsoft Azure RemoteApp was a Microsoft Azure cloud service that provides end users with access to Windows applications from any Windows, iOS, Mac OS X or Android device that has an Internet connection.

Unlike a regular RDS deployment, Azure RemoteApp used stateless servers which might be deleted/removed for instance during patching. Each user had a 50 GB user profile disk which is used to store data, the user could as well use other storage options like OneDrive/Dropbox etc..

Microsoft Azure RemoteApp Discontinued … on August 12, 2016 Microsoft announced that they were beginning the process of discontinuing their *Azure RemoteApp* service in favor of virtualized application and desktop services provided by their partner Citrix. (XenApp)

Microsoft Azure RemoteApp ---> service discontinued

RDS/RemoteApp ---> technology or platform and still very much alive!

RDS/RemoteApps – What is it?

Remote Desktop Services (RDS)

Platform that delivers individual virtualized applications, providing secure mobile and remote desktop access, and providing end users the ability to run their applications and desktops from the cloud.

Thin client environment – all the processing power is on the server.

Common solutions:

- Full desktop experience … user logs in and gets a desktop with the prepared Windows environment
- **RemoteApps** … individual applications that are hosted/run on the virtualized machine but appear as if they're running on the user's desktop like local applications

Reference Articles:
- https://www.vroege.biz/?p=2359
- https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/welcome-to-rds

RDS/RemoteApps Considerations

Infrastructure

- Where and how do I set this up?
- What versions of Windows and Access will be available to me?
- Can I set up multiple clients?

Costs

- Are there costs to this solution?
- What type of costs will I expect (startup, ongoing)?

Secuity

- How can I protect my solution from devious actors and my clients?
- Will I be able to protect my client’s data?

RemoteApps Infrastructure

As I stated on the title, this was a journey.

I am db developer, and I focus on making user friendly Access DBs. I had a steep learning curve on many topics:

- Azure,
- server architecture,
- server OS installation and setup,
- Active Directory and Group Policy Management
- licensing,
- scripting, PowerShell.

I had to make sure my solution met my clients needs, but as well fit my needs for a manageable solution with Access for me as the developer and ultimately the maintainer of the solution.

RemoteApps Infrastructure

RDP/Remote App requires a server and must meet cost, security, and developer experience considerations. And, of course, it must run the Access application and provide the user a smooth experience.

Options:

- Local server, personally managed ... mainly developer experience considerations
- Hosted solution through current business vendor (Newtek) or other hosting groups (Citrix) … mainly cost considerations
- Virtual Machine (VM) through Microsoft Azure … streamlined setup, scalable, hosted, monthly subscriptions that adjust to usage, servers that run current OS versions, menu-based options for security, other manged settings in Azure

Microsoft Azure seemed to meet all or most of the requirements.

RemoteApps Infrastructure

Most of my configuration was “on the fly”, as I had to learn things about the options for using the Access App remotely in order to configure the solution appropriately.

Setup included:

- Azure VM setup
- Administrative setup on the VM
- Installing Software on the VM for the RemoteApp setup, managing the server, and running the Access App.
- Active Directory for client users
- Remote Server Administration and Remote Desktop Licensing

RemoteApps Infrastructure

Key Software for the VM

1. Edge or Chome … server OS versions do not come with browser, or have IE.
2. Office … This is needed to run Excel, allow users to utilize their 365 accounts.
3. Access Runtime … Version of Access to use in my solution. Runtime version secures solution integrity.
4. RemoteApp Tool … Kim Knight's tool helps you create and manage RemoteApps and create connection files for each client.
5. PowerShell … comes with Windows 10, just make sure it is accessible.
6. Any software you need for your Access solution … Adobe Acrobat Reader DC for reports.

RDS/RemoteApps Considerations

Configuring the VM to include the Access app in a manageable structure that provides clients with a walled experience (“My” solution)

Each client will have their own database (accdb) file.

Why?

- Already structured this way in the current solution (run locally on their network), so there is a low bar to implement.
- Good option to “wall off” clients since app does not include true sign-in form that would filter out other data (although this is a future development area).
- Clients show hesitation to have data in a shared area anyway.

RDS/RemoteApps Considerations (cont.)

Each user would have their own front end file for their own unique connection to the server.

Why?

- Cardinal rule of Access Apps – do not have users open the same front end file.
- Users are already configured on the remote server with a separate login and user folder.
- The RemoteApp tool allows for configuration of user-specific connections.

RDS/RemoteApps Considerations (cont.)

My Access App will use a Runtime Version of Access.

Why?

- Further limits what users can do (or more to the point undo) with your applicaiton.
- Integral part of limiting devious outside users from accessing your app (more on this later).

Note : An important limitation to using Access Runtime is that you cannot develop your app on the VM because you cannot open the accdb version of the app on the VM. Development has to occur locally on your development machine; then publish to your VM environment.

RemoteApps Costs

Startup – mostly your time

- Knowledge … Learning about the ins and outs of RDS/RemoteApp (and any other things you don’t know, like Active Directory, Windows server, etc).
- Developing a solution … the cost for you to take an imagine the existing solution as a remote app solution
- Implementation … the cost for you to actually migrate the app over to the VM (setup the VM, setup users, licenses, etc.)

Startup – purchases

- RDP CAL licenses … required for any remote user on your VM, one-time purchase, these can be purchased in bulk at reduced pricing
- MS Office … depending upon your configuration, you may be able to use existing licenses, clients in this example bring their own license (federation?)

RemoteApps Costs

Ongoing - your time

- Managing the solution … using scripts to add users, installing Windows and other updates on the VM, shutting down the VM (to save costs)
- Managing your clients … sending them appropriate login information, updating when necessary

Ongoing - purchases

- Microsoft Azure … monthly “pay-as-you-go” (no prepayment option that makes sense), depends upon what specs you choose for your VM based upon your solution requirements

RemoteApps Security

In restricting a remote desktop user to a single application, I am also focused on security.

Security is rarely ever absolute. There are a few areas that were important in creating a solution that offers clients and users a secure environment by focusing on limiting the following:

- User access to one application on the VM
- Any user access within the VM to their own folders on the VM
- Client access to their only their own data
- Nefarious entry to the VM from client users or other outside actors

RemoteApps Security (cont.)

User access to one application on the VM

The focus was to limit what the user could access when opening the App. RemoteApp offers a way to serve up a focused user experience, and 99.7% of users want nothing more than what they are offered. For the cases where the user has the curiosity of a cat, limiting their capabilities becomes important to the safety of data and your application.

Using Active Directory Domain Services, setting all client users in one Organizational Unit (OU), then using the Group Policy Manager to configure the OU Start program (Access Runtime) as well as permitting the OU to use Remote Desktop will allow you to both allow users to run your App and limit them from using any other program on that Remote Desktop.

The **key** here is setting **Access Runtime** as the Start Program as the Remote Desktop Session Environment. Since your Access app opened via the RemoteApp connection will use Access Runtime, your App will open as expected. Any other application will have this same Access Runtime requirement (an important limitation that will help secure the solution).

Reference Articles:
- https://stackoverflow.com/questions/63022048/how-do-i-restrict-a-remote-desktop-user-to-a-single-application-on-windows-serve

RemoteApps Security (cont.)

Limit User access to their own folder and client access to their own data on the VM

After some trial and error in planning the setup of the clients and users on the VM, I determined what I needed to do to restrict user access certain folders.

Umbrella Permissions … Each user is a part of the RDS Users group, so I can set some permissions for all users by addressing permissions on the RDS Users Group (UAD top folder - DB, Documents).

Data Permissions … Each user for a client only has permissions to their specific client data folder

App Permissions … Each user has sole permission to their app folder

I use the menus in Windows folder Properties/Security menu to adjust user permissions and inheritance, then I can do an overall check of permissions using the AccessEnum program (part of the Windows Sysinternals utilities).

RemoteApps Security (cont.)

Dealing with devious actors

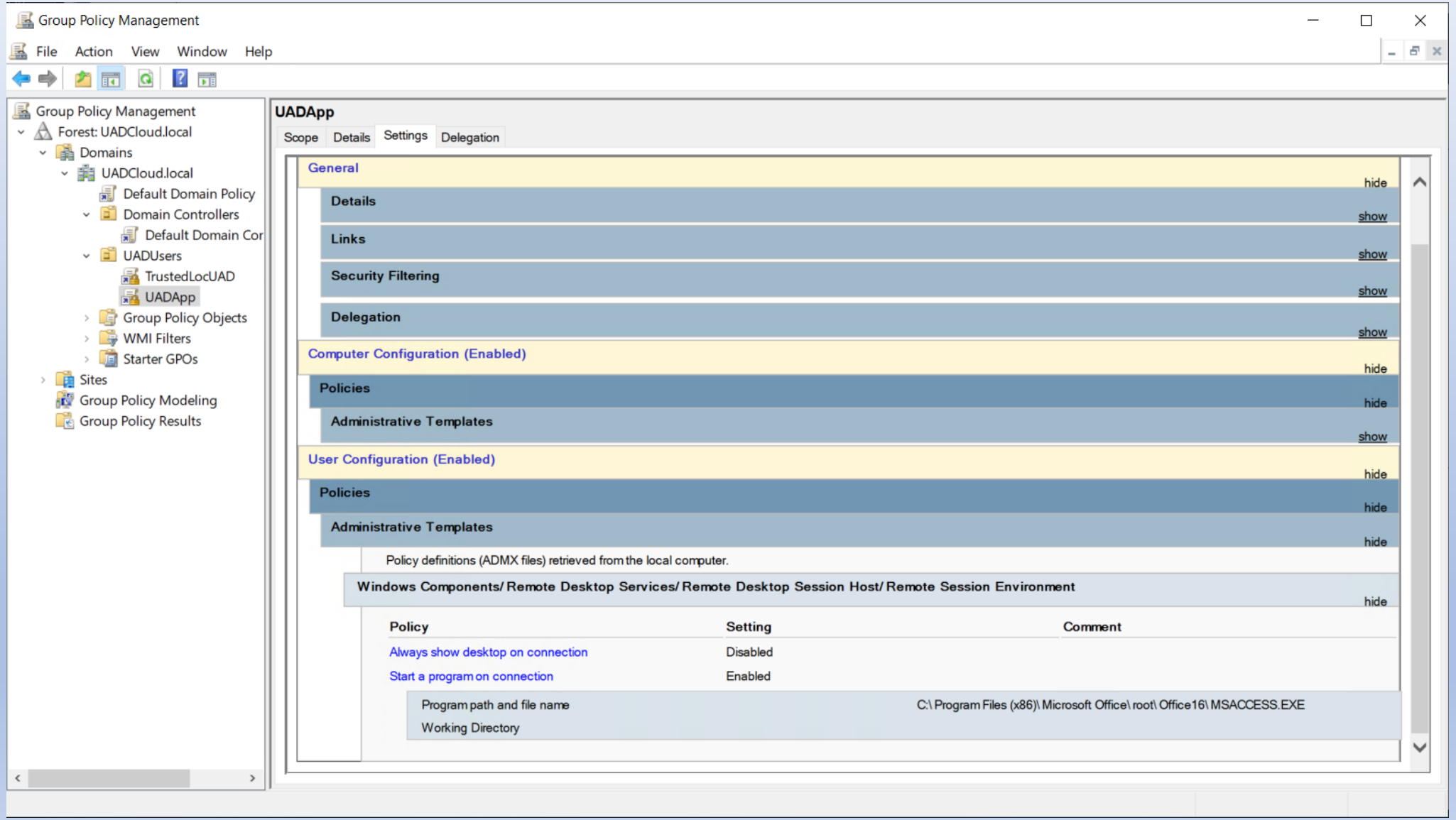
Not the Hollywood type – the people (clients or outside groups) who would attempt to connect to the VM to do nefarious deeds (make it a zombie, take down the server, ?).

One of security vulnerabilities in this solution is the RDP connection file created with the RemoteApp tool and provided to each user as their shortcut to the App. With a slight change to the **remoteapplicationmode** value (from 1 = RemoteApp, 0 = opens as desktop), the user or anyone with the RDP connection file and password could access the desktop directly.

In order to avoid having users open the connection directly to the desktop view, I restricted the remote desktop to a single application – Access Runtime. This limitation freezes out the user if they change the **remoteapplicationmode** value to 0 (desktop).

Reference Articles:
- https://www.donkz.nl/overview-rdp-file-settings/

RemoteApps Security (cont.)



RemoteApps Security (cont.)

Dealing with devious actors (cont.)

Azure has settings as well that will assist you with setting security on the solution.

One particular setting on the is the Networking / Inbound Port rules. Your clients may have a particular source IP address. You can then set the inbound rule that the VM only accept inbound communication from a particular port.

As well, you can set the port that will accept communication from inbound sources. Using a slightly different port number, not 80 (internet traffic) and not 3389 (commonly used for RDP traffic). For port 3389, I only allow inbound traffic from my developer PC IP address.

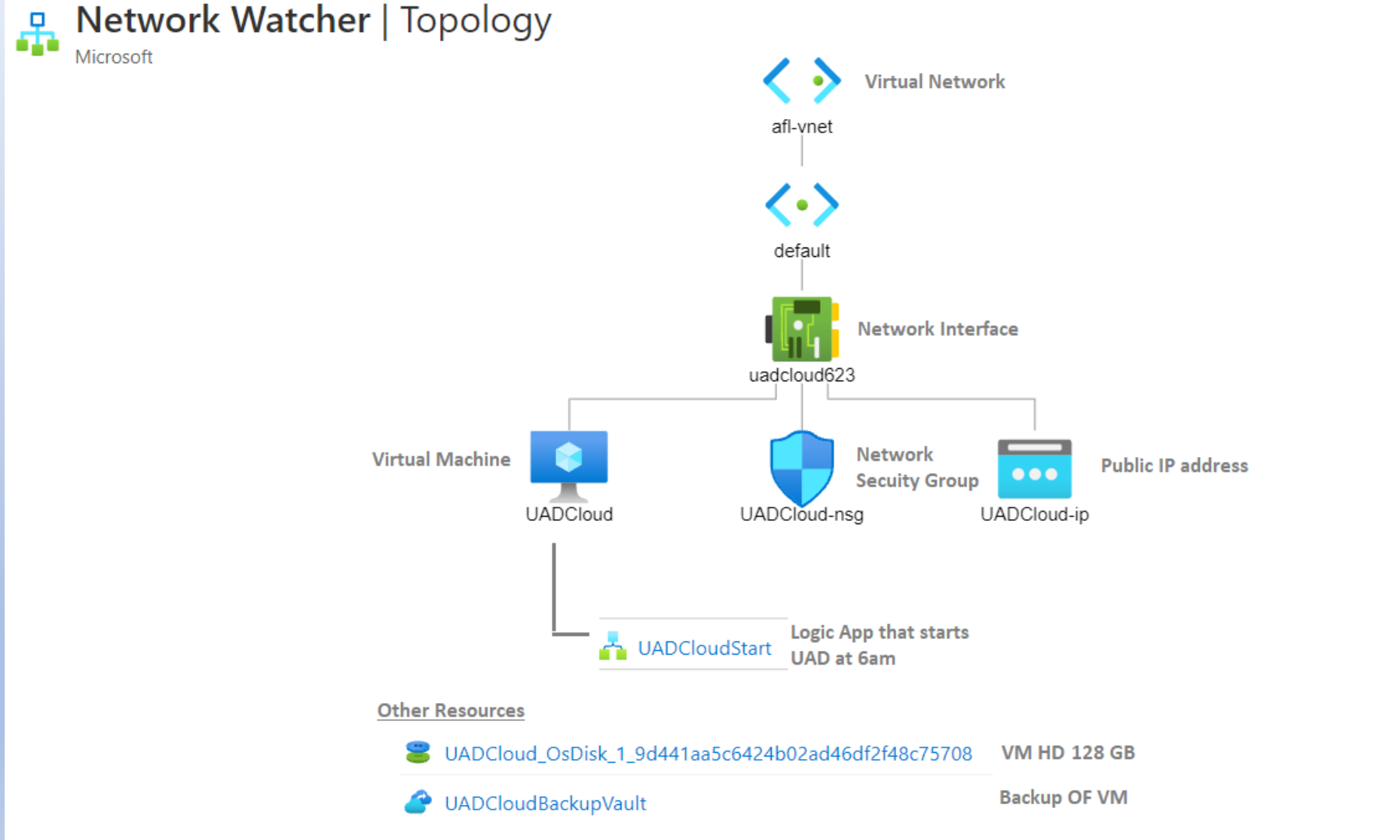
Since I have multiple clients that will be inbound on multiple IP addresses, I've done what I could to limit port scanning / sniffing by using an uncommon port number for client RDP access. The RDP connection string will use this port number as the server port in the string setting.

Specifications for Remote Solution

VM and App Details

- Azure VM
  - B-series, General Purpose, 2 CPUs, 8 GB RAM, 128 GB HD ($65-70 / month)
  - 4 Data disks (I only use 1)
  - Windows Server 2019 Datacenter
- UAD
  - 2016 Access, 64-bit OS, 32-bit Office
  - Front End, 60-70 MB, 12 users
  - Back End, 15-20 MB, 3 clients

Azure Diagram of Remote Solution



Active Directory/ GPO Setup

Domain
➔ UADCloud.local

Organizational Unit (OU)
➔ UADUsers
All UAD users will be added to this OU

GPO
➔ TrustedLocUAD
Policy - Trusted Location #1 ... C:\Users (UAD User Front End Location)
Policy - Trusted Location #1 ... C:\UAD\DB (UAD User data location)
UADUsers OU linked to this policy
➔ UADApp
Both Computer and User configuration
Always show desktop on connection ... **Disabled**
Start a program on connection ... **Enabled**
Program Path - C:\Program Files (x86)\Microsoft Office\root\Office16\MSACCESS.EXE
UADUsers OU linked to this policy

Trials, Tribulations, and Foibles

This solution took many hours of my learning / development time, and it was not without some drama….

- Locking out Admin Account of VM, freezing me out of initial solution development

  Lesson learned – Make sure that the Admin account is not in the OU that includes the remote users, and of couse, have a backup

- Azure support experience, staff is sometimes not knowledgeable.

  Lesson learned – learn as much as you can, can't rely upon support to be the "expert"

- SSL Certificate cannot be applied to VM (I do not own that IP address)

  Lesson learned – take victories where you can and realize that no solution is perfect

Reference Articles:
- https://sectigostore.com/page/ssl-certificate-for-ip-address/

Conclusion

Not for the faint of heart … but doable

Preplan ! … make sure this will work for your solution before you undertake it

Some “Thank Yous!” to my helpers during the journey…

- Kim Knight (RemoteApp Tool),
- Luke Chung (Presentation on RemoteApp as a solution)
- Cary Jensen (Awesome Stackoverflow article on restricting RD users to a single app)
- George Young (Reviewing my remote app solution for sanity purposes)
- Bill Kindle (article on NTFS Permissions With PowerShell)
- Daniel Pineault (random string generator used for setting up RDP users)
- Microsoft Azure Support (Navigating Azure and helping me through the VM lockout

Conclusion