

# Windows VDI Client to Windows Virtual Desktop (WVD) Migration: Technical Guide

**MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.**

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation. Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, our provision of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property. The descriptions of other companies' products in this document, if any, are provided only as a convenience to you. Any such references should not be considered an endorsement or support by Microsoft. Microsoft cannot guarantee their accuracy, and the products may change over time. Also, the descriptions are intended as brief highlights to aid understanding, rather than as thorough coverage. For authoritative descriptions of these products, please consult their respective manufacturers. © 2016 Microsoft Corporation. All rights reserved. Any use or distribution of these materials without express authorization of Microsoft Corp. is strictly prohibited. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

## Contents

1. Introduction .....	5
2. Target Audience .....	5
3. Prerequisites/Requirements.....	5
4. RDS Setup (on-premises).....	7
5. Migration Workflow.....	9
6. Planning & Design.....	10
6.1. General Best Practices.....	10
6.2. Discovery & Assessment of on-prem RDS infrastructure .....	11
6.2.1. Access & Export the Assessment Results.....	11
6.2.2. WVD Session Host VM & Storage SKU Guidance.....	13
6.3. Azure networking .....	14
6.4. Identity & Access Management.....	16
6.5. Azure Storage and Disks.....	21
7. Implement a WVD PoC (Proof of Concept) .....	21
7.1. Grant Azure Active Directory permissions to the Windows Virtual Desktop service .....	22
7.2. Assign the Tenant Creator Application role to a user in your Azure Active Directory.....	22
7.3. Download the WVD PowerShell Module .....	23
7.4. Create the WVD Tenant.....	23
7.5. Create Host Pools using ARM template .....	25
7.6. Manage App Groups .....	28
7.7. Validate user connections to WVD.....	31
8. Migrate on-prem VDI based RDS resources to Azure-WVD.....	35
8.1. Lift-n-Shift to Azure - Detailed Migration Steps.....	36
8.1.1. Hyper-V .....	36
8.1.2. VMWare.....	37

9. Appendix.....	38
9.1    Deploy WVD Session Hosts using custom Image.....	38
9.2    Deploy WVD Session Hosts using custom VHD in blob storage .....	41
9.3    Install WVD Agents manually .....	44
9.4    Check Group Policy updates remotely.....	48
10. Support.....	49

## 1. Introduction

The ***primary intent of this guide is to illustrate transitioning from an on-premise VDI based RDS deployment (Client OS) to WVD Session Host (Client OS) in Azure.*** It is intended to be used by Customer & Partners to help familiarize themselves with the processes, methodologies and tools required to migrate their on-premise workloads into Azure and integrate with the WVD service/platform.

## 2. Target Audience

This document is ***Level 400+ technical migration guide*** primarily intended for Azure Specialists, Cloud Solution Architects, Migration experts, System Administrators & anyone else who are going to be hands-on in executing the on-premise to Azure (WVD) migrations. It is assumed that the audience has deep insights into their on-premise workload architectures, storage & networking capabilities along with the interdependencies across multiple services/components involved like Active Directory, VDI based RDS deployments, Microsoft Azure and its core services (compute, storage & Network).

Please note that this document will primarily focus on the detailed migration process and is NOT a primer for the technologies afore mentioned.

## 3. Prerequisites/Requirements

Before getting started, **all** items listed below **must** be checked/validated to ensure the most basic requirements are in place to proceed with executing the remaining steps in this guide. ***For any reason, if you do NOT meet all requirements, then either get the required access or get in touch with a team/person who can help you achieve the same.***

- ✓ Presence of an on-premises datacenter / dev-test environment where you currently have active VDI based RDS deployments.
- ✓ Need admin level access and in-depth knowledge to manage infrastructure level services like Active Directory, DNS, Networking etc.
- ✓ Admin level access to all the components in the on-premises RDS deployments is required along with in-depth knowledge of how RDS deployments are configured, user access, ETC.
- ✓ Knowledge and comfortability in managing Azure services like:
  - Azure Compute (VMs/Availability Sets)

- Azure Storage (disks/storage accounts)
- Azure Networking (VNET/Subnets/NIC/NSGs)
- Azure Migrate & Azure Site Recovery
- Azure Active Directory & Azure AD Connect
- ✓ An Azure tenant (Ex: yourdomain.onmicrosoft.com) environment along with at least 1 active subscription.
  - If you are a customer, then reach out to your CSP partner who can provide you with the required tenant information and access
  - If you are the CSP partner, then you can get the customer details by logging onto the [Microsoft Partner Portal](#) > dashboard > customers . Here you can see the domain under the column Primary Domain Name
- ✓ Ensure that the user who will provision & configure WVD must have "Global Admin" rights to the Azure tenant they are a part of.
  - Based on the operating model, some customers might not have this enabled so contact your Partner who can help with the same.
- ✓ Ensure that the user who will provision & configure WVD must have at least "Contributor" rights to the Azure subscription
  - Based on the operating model, some customers might not have this enabled so contact your Partner who can help with the same.
- ✓ A Windows Server Active Directory environment (minimum 1 domain controller + DNS server) must be present on-premises.
  - Eventually this will be sync-ed with Azure Active Directory using Azure AD Connect.
- ✓ A S2S (Site to Site) IPSEC tunnel OR Express Route will be required to extend your on-premises network into azure.
  - Make sure you have a compatible VPN device and someone who can configure it. For more information about compatible VPN devices and device configuration, see [About VPN Devices](#).
  - Verify that you have an externally facing public IPv4 address for your VPN device.
  - If you are unfamiliar with the IP address ranges located in your on-premises network configuration, you need to coordinate with someone who can provide those details for you. When you create this configuration, you must specify the IP address range prefixes that Azure will route to your on-premises location. None of the subnets of your on-premises network can overlap with the virtual network subnets that you want to connect to.

- If planning to setup Express Route, the prerequisite checklist can be found [here](#)
- ✓ Ability to work with command line implementation using PowerShell, Azure Modules.
- ✓ The ability to manage ARM templates and deploy azure resources with it.
- ✓ The [WVD requirements](#) must be satisfied.

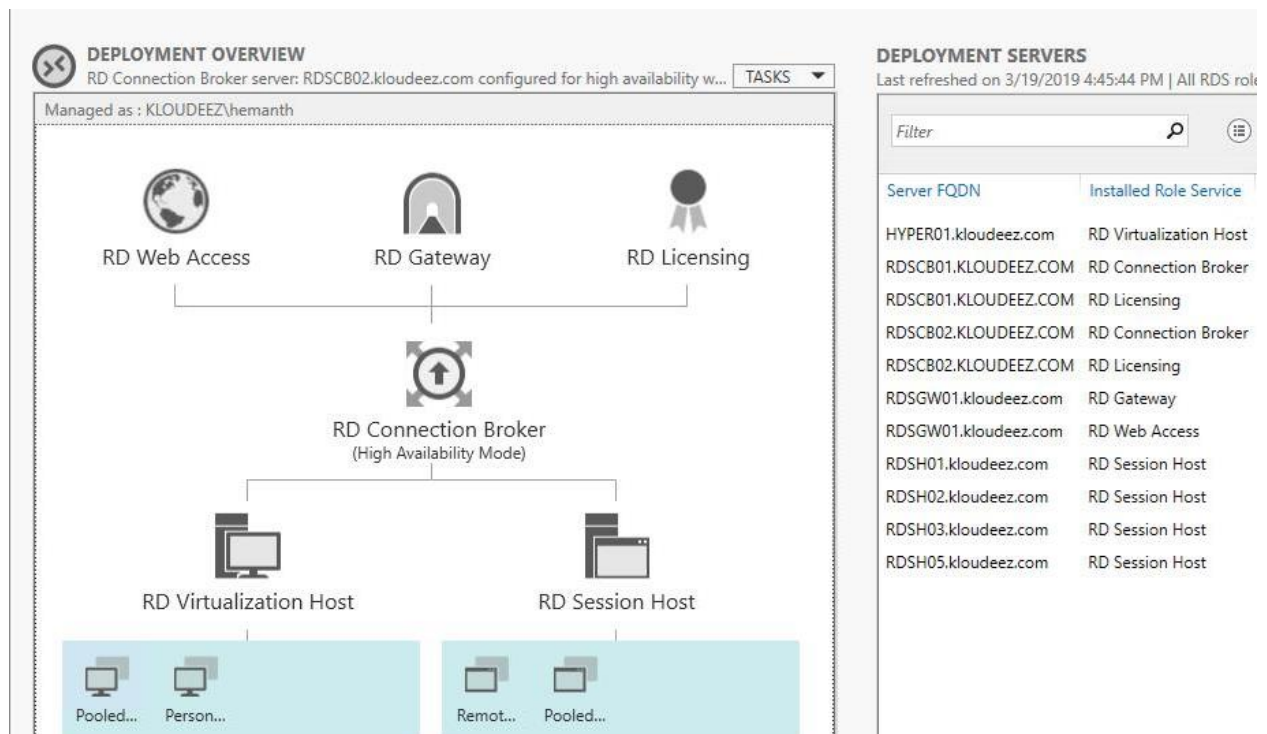
NOTE:

- Always open PowerShell in administrator mode
- The screenshots in this document are for reference only. There might be instances where the instructions and the details in the screenshot are different so, **\*always ensure to clearly read & follow the instructions\***

## 4. RDS Setup (on-premises)

This section describes a **typical RDS on-premises configuration** to simulate an existing enterprise architecture setup that will eventually be migrated to Azure. Please be advised this baseline configuration is to help understand/summarize the standard migration process and your enterprises` implementation may contain additional customizations and/or settings that **might not** be covered in this document.

Below is the on-premises RDS deployment overview and the breakdown of the different server roles being used in the environment.



*On-premises RDS Deployment*

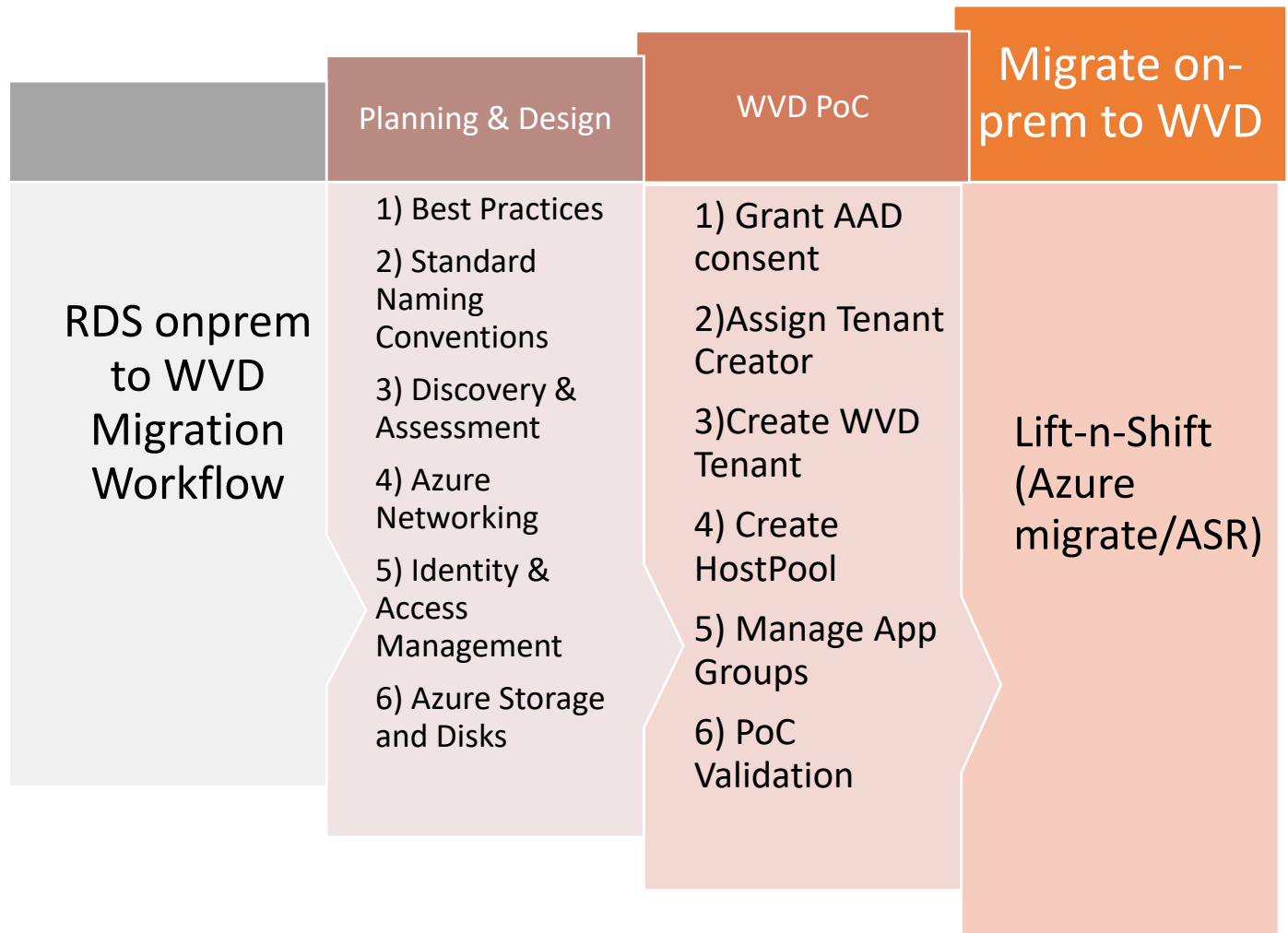
- Active Directory
  - All VMs are domain joined. For the sake of this document, we are using a fictitious single forest/domain called Kloudeez.com (*similar to contoso.com OR fabrikam.com references found in any Microsoft public documentation*)
  - There is a single domain controller called ADC01 that also doubles up as the DNS server.
- Connection Broker
  - 2 connection brokers RDSCB01 & RDSCB02 are configured in a highly available (HA) configuration
  - The broker configuration data is stored on a standalone SQL server RDSSQL01 that both brokers can access.
- RD Gateway & Web Access
  - A standalone server RDSGW01 has both the RD-Gateway and Web Access roles installed
- RD-Licensing
  - The RD-licensing role is also installed on RDSCB01 & RDSCB02 (connection broker)



- The licensing mode configured is per-user
- RD-Session Hosts
  - We have a total of 4 Session hosts servers (RDSH01/02/03/05)
  - These are the servers where session-based RemoteApp & RemoteDesktops (pooled) collections are hosted.
- RD-Virtualization Host (**Focus of this Document**)
  - A Hyper-V host HYPER01 is being used for VM based collections.
  - This Host will deliver Windows client based Personal Desktops to users (1:1).
- File Server
  - A standalone file server called RDSFS01 accessible vis SMB share endpoint to store profile data using UPD (User Profile Disks).
  - **Note:** We don't need this for VDI Based Windows 10 Personal Desktops.

## 5. Migration Workflow

This section summarizes the different phases involved in migrating the on-prem VDI based RDS environment to WVD. Typically, there are some preliminary steps, common across ALL migration approaches and the actual migration steps are split into different models based on the customer's WVD requirements. The below graphic should provide a 100FT overview of the different phases involved that are described in the latter sections.



*WVD Migration Workflow*

## 6. Planning & Design

### 6.1. General Best Practices

Since everyone's business and technical requirements vary across the board, it is always a good idea to familiarize yourselves with the standard best practices across the different Azure technologies & services.

- Standard naming Conventions
  - Skip this section if you are already following a standard naming convention for resources on-prem and in Azure. If not, please follow the guidance in the [link](#) to maintain a consistent naming convention across your resources

- [Azure security best practices and patterns](#)
  - Azure Networking & security [Best Practices](#)
  - [Implementing a secure hybrid network architecture in Azure](#)
  - [Best practices for Azure VM security](#)
- Azure Active Directory Hybrid Identity [best practices](#)
  - [Azure identity management and access control security best practices](#)
- Azure Storage security [overview](#)

## 6.2. Discovery & Assessment of on-prem RDS infrastructure

The Azure Migrate service assesses on-premises workloads for migration to Azure. The service assesses the migration suitability of on-premises machines, performs performance-based sizing, and provides cost estimations for running on-premises machines in Azure. If you're contemplating lift-and-shift migrations, or are in the early assessment stages of migration, this service is for you.

In case you already have the Azure Total Cost Ownership (TCO) and/or the azure VM SKU requirements for your WVD infrastructure finalized, then skip this section all together.

The steps below will provide guidance on how to get started with a quick assessment of your existing RDS/VDI infrastructure, download the assessment report and convert those details into a meaningful plan for your WVD planning.

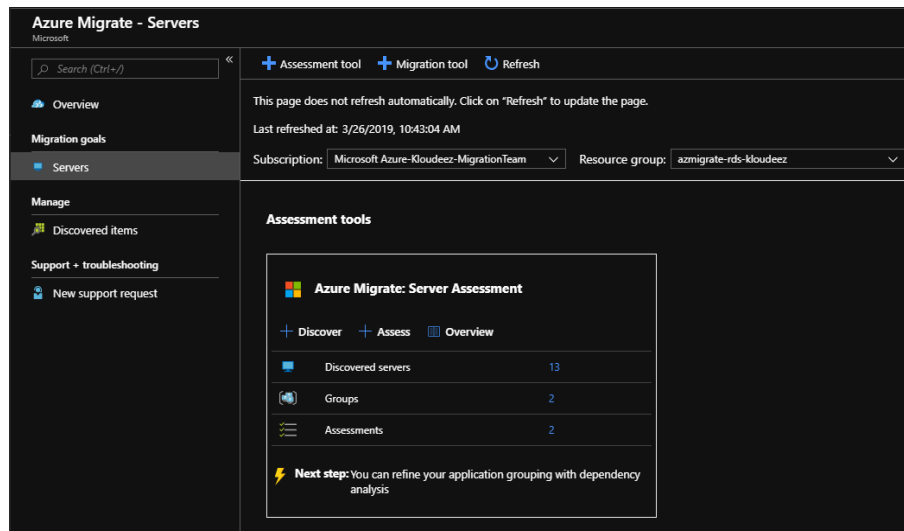
- Start with the [Azure migrate-overview](#) to generally understand the product and it's requirements.
- Based on the Hypervisor infrastructure being used on-prem, choose the respective option to deploy the Azure migrate project and start the assessment.
  - [Assess VMware VMs](#) - If your infrastructure operates on VMware
  - [Assess Hyper-V VMs](#) - If your infrastructure operates on Hyper-V.

*Please note that this feature is in preview.*

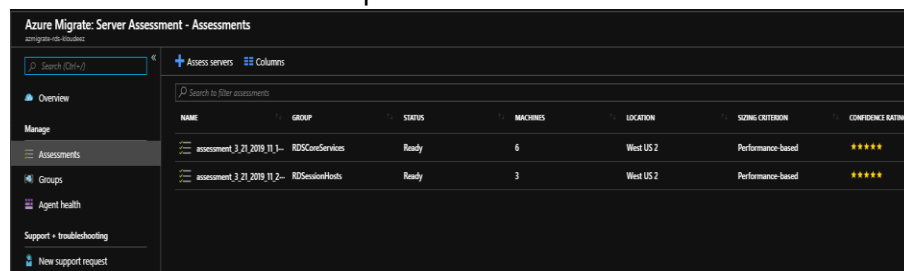
### 6.2.1. Access & Export the Assessment Results

Once your assessment results are ready, follow the instructions below on how to use that assessment data and plan for your WVD infrastructure.

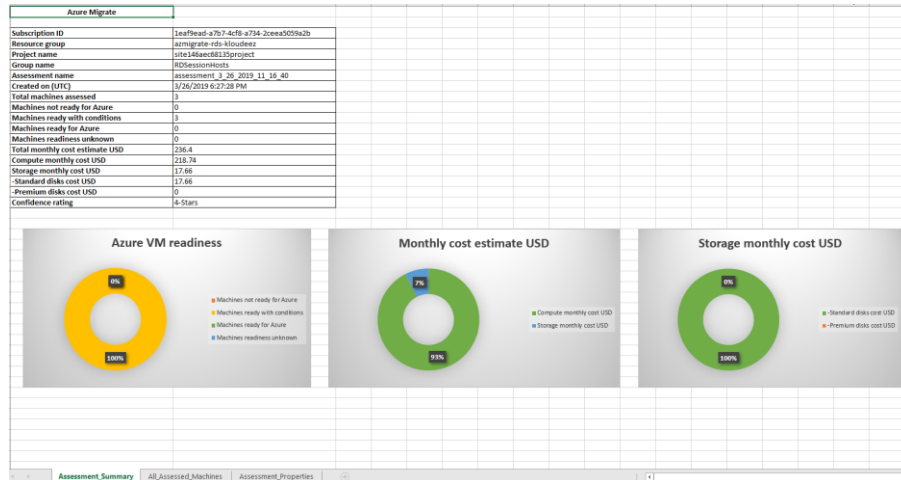
1. Export the Assessment Data. Open the Azure Migrate Overview through the Private Preview link and select the appropriate resource group to get the Assessment tools.



2. Click on the Assessments to open them.



3. Click on the assessment name to view and click on Export assessment to download as an Excel.
4. The first sheet in the assessment lists out the Azure TCO calculations for running these VMs in Azure. *This is an FYI and if you need to modify and adjust the TCO to a desired \$ then please follow the Azure Migrate documentation links shared earlier in this section.*



5. Navigate to the All\_assessed\_Machines sheet to view the recommended VM SKUs and Disk sizing based on the Assessment.
  - i. Recommended Azure VM SKUs
  - ii. Memory & CPU details
  - iii. Disk sizing & SKU classification (Standard Vs Premium)

## 6.2.2. WVD Session Host VM & Storage SKU Guidance

Based on your end goals & requirements, the planning & selection of the WVD session host VM SKUs can be done in a couple of different ways as outlined below.

1. Use the VM SKU & storage recommendations by Azure Migrate in the Assessment as seen below.

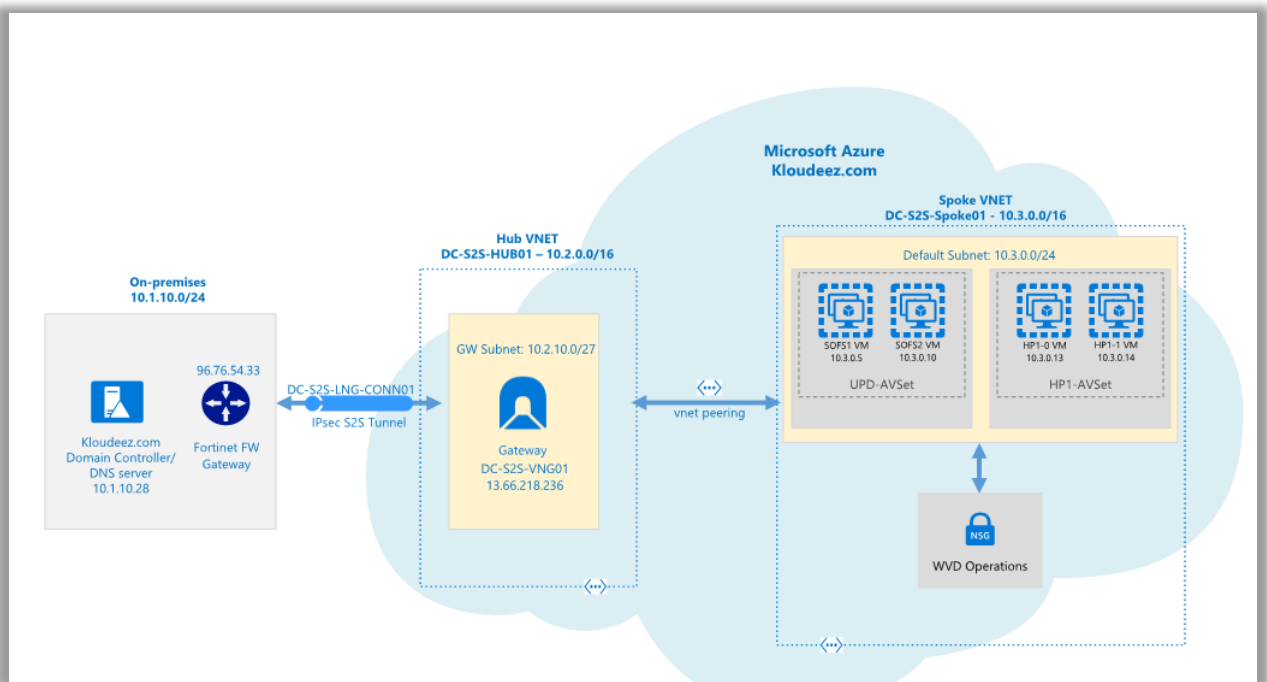
Machine	Recommended size	Operating system	Cores	Memory(MB)	Storage(GB)	Standard disks	Premium disks	Network adapters	IP address	MAC address	Network in(Mbps)	Network out(Mbps)
Personal-0	Standard_D1_v2	Windows 10 Enterprise Evaluation	2	8192	127	1	Not applicable	1	[10.1.10.53,fe80::95d7:79b9:3a78:cc95];	[00:15:5d:0a:de:0f];	0.01	0.01
Personal-1	Standard_D1_v2	Windows 10 Enterprise Evaluation	2	8192	127	1	Not applicable	1	[10.1.10.54,fe80::7cbe:a0f:d66a:1fab];	[00:15:5d:0a:de:10];	0.03	0.01

2. Depending on the customer's flexibility, they can also choose a SKU that better suits their needs. Such as a Compute Optimized vs Memory Optimized vs High Performance Compute etc.

### 6.3. Azure networking

The recommendation is to design your Azure Networking using a [Hub-Spoke topology](#). Consider the HUB like a DMZ deployed with your Virtual network Gateways and other security/edge appliances like Firewalls Etc. while the Spoke will act as the backend zone where your session hosts servers are deployed to and is peered with the HUB.

Below is the architecture diagram that outlines the Azure Networking plan that was deployed for the sake of this migration guide.



WVD Network Architecture

The sections below will briefly summarize the components deployed as a part of the Azure networking plan. ***It is \*highly recommended\* that your networking team is consulted during this phase for an optimal implementation.***

### **1.1. Azure Virtual Networks (VNET)**

Like discussed earlier we are going to create 2 VNETs in a HUB and Spoke model using the below details.

#### **1.1.1. HUB**

- VNET Name: DC-S2S-Hub01
- CIDR: 10.2.0.0/16 *or anything else which does not overlap/conflict with any existing networks*
- Create a subnet called "GatewaySubnet" (this is cannot change and will host the Virtual Network Gateway)
- Based on your requirements, choose an Azure virtual Network Gateway using the specifications from [Gateway SKU](#) and deploy it to the "GatewaySubnet"

#### **1.1.2. Spoke**

- VNET Name: DC-S2S-Spoke01
- CIDR: 10.3.0.0/16 *or anything else which does not overlap/conflict with any existing networks.*
- Create a subnet called "Default" (or make it specific based on how you want to isolate & manage servers)

#### **1.1.3. VNET Peering**

- Configure Peering across the HUB & Spoke VNETs so that resources in networks Hub (10.2.0.0/16) & Spoke (10.3.0.0/16) can communicate with each other.

#### **1.1.4. S2S Connectivity between on-premises & Azure.**

Please consult your networking team to understand and implement steps in this section.

- Based on your bandwidth, latency & security requirements first choose between the connectivity model.

- S2S or Express Route. *[For the sake of this document, we will be using S2S IPSEC tunnel]*
- Follow the instructions below to build an S2S-IPSEC tunnel using the on-premises edge networking device.
  - Read through the [vpn-gateway, Bandwidth requirements](#) to finalize your requirements first
  - [Create the VPN Gateway](#)
    - *Like shown in the above diagram, an Azure virtual network gateway called "DC-S2S-VNG01" has been deployed to the HUB VNET and a static publicIP address 13.66.218.236 assigned to it.*
  - Use the instructions at [Build an S2S IPSEC tunnel with Azure](#) and complete the connectivity to azure.
    - *From the architecture diagram, a connection "DC-S2S-LNG-CONN01" back to the on-premises device (96.76.54.33) has been created in Azure.*
  - Update the VNET with your on-premises DNS servers using the instructions at [Change DNS servers](#)
    - *For both the HUB & SPOKE VNETs the DNS servers has been updated to 10.1.10.28(on-prem). IF you are planning to deploy additional Domain Controllers in Azure, please remember to add those as well once ready.*
  - Now you should be able to launch a VM in the Spoke VNET > domain join and access it like a local resource.

## 6.4. Identity & Access Management

This section will cover a multitude of areas starting for provisioning AD security groups & organizing users, creating GPO objects, extending your Identity into Azure ETC. It is highly recommended to work with your AD team for this section.

It is very important to review and choose the correct hybrid identity for your organization using the guidance at [What is hybrid identity?](#) . For the sake of this document, we are going to choose Azure AD Connect

### 1.2.Create Test Users and AD Security Groups



For the sake of implementing a WVD PoC, we will be creating some test users' objects & AD Security groups that can be used to validate WVD functionality without disrupting everyday operations.

1. Let's start by creating some test users that will later be used to grant access to remote desktops & apps.
2. Log onto the domain controller > open PowerShell and run the below command

```
#update values first
$name = "VDIUser1"
$UPN = $name + "@yourdomain.com"
$pass = ("Passme1!" | ConvertTo-SecureString -AsPlainText -force )
```

```
Import-Module ActiveDirectory
```

```
New-ADUser -UserPrincipalName $UPN -AccountPassword $pass -
DisplayName $name -Name $name -ChangePasswordAtLogon $false -
PasswordNeverExpires $true -Enabled $true
```

```
PS C:\WINDOWS\system32> $name = "VDIUser1"
PS C:\WINDOWS\system32> $UPN = $name + "@yourdomain.com"
PS C:\WINDOWS\system32> $pass = ("Passme1!" | ConvertTo-SecureString -AsPlainText -force )
PS C:\WINDOWS\system32> New-ADUser -UserPrincipalName $UPN -AccountPassword $pass -DisplayName $name -Name $name -ChangePasswordAtLogon $false -PasswordNeverExpires $true -Enabled $true
```

Update the values and Repeat the command for as many users you like to test with.

3. Now let's create the Security group(s) that will be required to manage resources and grant access at different stages in the subsequent sections. Below is a list of the security groups we need and why.
- 4.

SecurityGroupName	Description
VDI-DesktopUsers	Contains users that need access to Remote Desktops hosted using WVD

Execute below commands on the domain controller in PowerShell

```
#update values using the first
```

```
$SecurityGroupName = "value from the SecurityGroupName column"
$Description = "value from the Description column"
```

```
New-ADGroup -Name $SecurityGroupName -SamAccountName
$SecurityGroupName -GroupCategory Security -GroupScope Global
- -Description $Description
```

```
h32> $SecurityGroupName = "TestGroup"
h32> $Description = "Group with Test Users"
h32> New-ADGroup -Name $SecurityGroupName -SamAccountName $SecurityGroupName -GroupCategory Security -GroupScope Global -Description $Description
```

Now let's add the test users to the Security Groups. Execute below commands on the domain controller in PowerShell

```
#adding users to RemoteDesktop group
```

```
$Identity = "VDI-DesktopUsers"
```

```
Add-ADGroupMember -Identity $Identity -Members
@("VDIuser1", "VDIuser2")
```

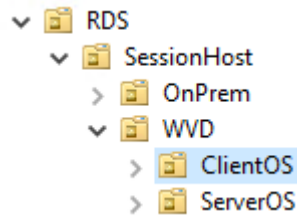
### 1.3.Active Directory Organization Unit (OU) structure for WVD session hosts.

It is strongly recommended to consult your in-house expert for this section. The below guidance is subjective, and every enterprise should have an already established process/guidelines to manage their AD computer objects. Consider the below information as a mere FYI to help understand the steps to setup a OU structure.

Since we are introducing new VMs into the existing environment and would most likely manage them using GPOs (Group Policy Objects), it is important to plan accordingly.

1. On your domain controller, open ADUC (dsa.msc)
2. Expand the domain and get to the RDS OU (consider this the main OU where all your on-prem RDS computer objects are stored)
  - Under RDS, create a sub OU called Session Host (or Likewise) to manage common settings for all session hosts (on-prem & WVD)
  - Under Session Host, create a sub OU called WVD (or Likewise) to manage the WVD session hosts

- Under WVD, create a sub OU called "ClientOS" (the idea is to store all session hosts that serve Windows Client Desktops relative to the purpose of your HostPool in WVD)



3. Open Active directory users & computers (dsa.msc)
4. Go to the Container/OU where your session hosts are present > CTRL + select the WVD session hosts > right click > move > and move them to the Respective WVD OU > Click OK

Note: Revisit this section and perform steps 3 and 4 when the session hosts are provisioned. Restart the VMs after adding them to an OU.

#### 1.4.Add Custom Domain in AAD

Every new Azure AD tenant comes with an initial domain name, **domainname.onmicrosoft.com**. You can't change or delete the initial domain name, but you can add your organization's names to the list. Adding custom domain names helps you to create user names that are familiar to your users, such as [alain@contoso.com](mailto:alain@contoso.com).

Skip this step if already complete or the steps below provide the guidance to complete the same.

- Start with the steps using [add-custom-domain](#)
- Once the above is completed, you can login to the Azure portal > AAD > Custom Domain Names > to see that your organization name (EX: domain.com) is set as primary

Search			
NAME	STATUS	FEDERATED	PRIMARY
kloudeez.com	✓ Verified		✓
qlbl.onmicrosoft.com	✓ Available		

### 1.5. Install & Configure Azure AD Connect

The on-premises infrastructure must already have an Active Directory (AD) environment. Before we start migrating any resources, it is important to Integrate your on-premises directories with Azure AD to make your users more productive by providing a common identity for accessing both cloud and on-premises resources.

The section below will provide guidance on the implementation of Azure AD Connect as a hybrid identity solution for your migrations.

- Understand the service using [whatis-azure-ad-connect](#)
- The pre-requisites & general requirements are explained using [Prerequisites for Azure AD Connect](#)
- Install Azure AD Connect using [how-to-connect-install-express](#)
- After the installation and an initial sync is complete, you should be able to see your on-prem user accounts and security groups show in Azure ( Azure Portal > AAD > Users OR Groups )

RD	RDSuser1	RDSuser1@kloudeez.com	Member	Windows Server AD
RD	RDSuser2	RDSuser2@kloudeez.com	Member	Windows Server AD

AC	AccessFSLogix	Security	Synced
AC	AccessUPDServers	Security	Synced

### 1.6. Extend on-prem AD into Azure

Since we already have a S2S IPSEC tunnel between the on-prem datacenter & Azure, we are NOT deploying additional domain controllers in Azure. This might be OK for implementing a WVD PoC although, for the actual migrations you would want to extend/deploy additional domain controllers in Azure before you start migrating any other resources to meet security & latency requirements.

- Please follow the guidance from [Extend Active Directory Domain Services \(AD DS\) to Azure](#)

## 6.5. Azure Storage and Disks

Please refer the storage assessment results to finalize the Azure storage / Disk options required for your VM's. As a baseline recommendation, it would be advised to choose the following for production workloads. We are NOT going to deploy anything at this time and will be handled in the subsequent sections.

- **Azure Storage Account**

Azure Storage account would be primarily be utilized for hosting the diagnostic logging information across ALL your VMs.

- Choose SKU = Standard for low-pri (Dev/Test) workloads/servers
- Choose SKU = Premium for High pri (Production) workloads/servers

- **Azure Disks**

For better management and efficiency, the recommended storage solution for all VMs in Azure must be **Managed Disks** and it is highly encouraged to avoid using **un-managed disks (blob VHD on a Storage Account)** unless there is a very strong business requirement.

- Choose SKU = Standard for low-pri (Dev/Test) workloads/servers
- Choose SKU = Premium for High pri (Production) workloads/servers

## 7. Implement a WVD PoC (Proof of Concept)

Creating a tenant in Windows Virtual Desktop is the first step towards building out your desktop virtualization solution. A tenant is a group of one or more host pools. Each host pool consists of multiple session hosts, running as virtual machines in Azure and registered to the Windows Virtual Desktop service. Each host pool also consists of one or more app groups that are used to publish remote desktop and remote application

resources to users. With a tenant, you can build out host pools, create app groups, assign users, and make connections through the service.

The subsequent sections will detail the step-step process to implement a working WVD solution in Azure.

## 7.1. Grant Azure Active Directory permissions to the Windows Virtual Desktop service

1. Open a browser and connect to the [Windows Virtual Desktop consent page](#).
2. For **Consent Option** > **Server App**, enter the Azure Active Directory tenant name or Directory ID (from the Azure portal), then select **Submit**.
  - For Cloud Solution Provider customers, the ID is the customer's Microsoft ID from the Partner Portal.
  - For Enterprise customers, the ID is located under **Azure Active Directory** > **Properties** > **Directory ID**.
3. Sign in to the [Windows Virtual Desktop consent page](#) with a global administrator account. For example, if you were with the Contoso organization, your account might be admin@contoso.com or [admin@contoso.onmicrosoft.com](#).
4. Select **Accept** > wait for one minute.
5. Navigate back to the [Windows Virtual Desktop consent page](#).
6. Go to **Consent Option** > **Client App**, enter the same Azure AD tenant name or Directory ID, then select **Submit**.
7. Sign in to the Windows Virtual Desktop consent page as global administrator like you did back in step 3. Select **Accept**.

## 7.2. Assign the Tenant Creator Application role to a user in your Azure Active Directory.

1. Open a browser and connect to the [Azure Portal](#) with your global administrator account.
  - a. If you're working with multiple Azure AD tenants, it's best practice to open a private browser session and copy and paste URLs into the address.
2. Select Enterprise applications, search for Windows Virtual Desktop and select the application.
3. Select Users and groups, then select Add user. *Ensure that this is either a service or a user account that does not have MFA/CA enabled*

4. Select Users and groups in the Add Assignment blade
5. Search for a user account that will create your Windows Virtual Desktop tenant.
  - a. For simplicity, this can be the global administrator account.
6. Select the user account, click the Select button, and then select Assign.

### 7.3.Download the WVD PowerShell Module

1. Download the Windows Virtual Desktop module and save the package in a known location on your computer. For example, C:\temp.
2. Find the downloaded package. Right-click the zip file, select Properties, select Unblock, then select OK. This will allow your system to trust the module.
3. Right-click the zip file, select Extract all..., choose a file location, then select Extract.
4. First, run this cmdlet to save the file location of the extracted .zip file into a variable:

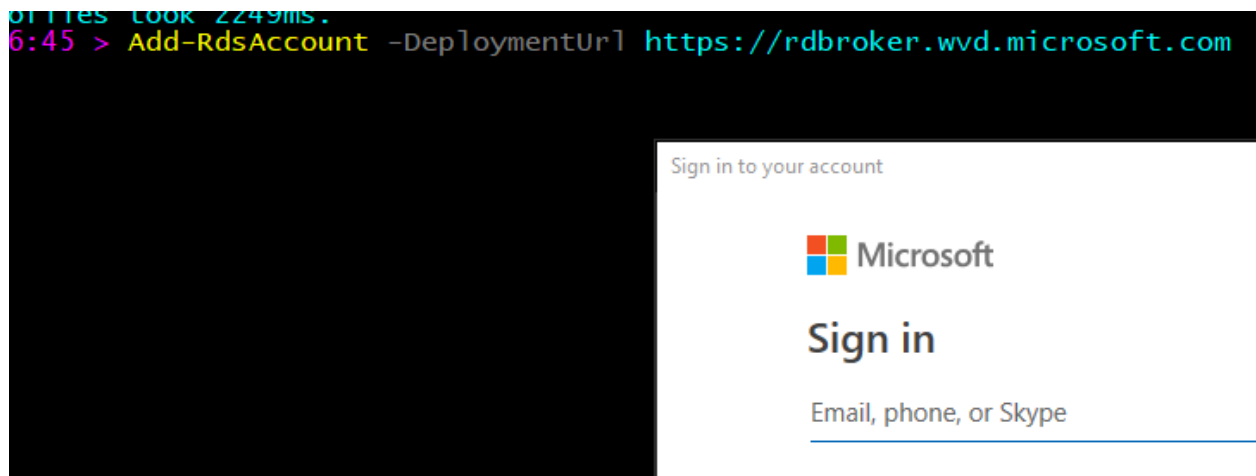
```
$module = "<extracted-module-location>"
```
5. Second, run this cmdlet to import the DLL for the module:

```
Import-Module $module\Microsoft.RDInfra.RDPowerShell.dll
```
6. You can now run Windows Virtual Desktop cmdlets in your PowerShell window. If you close your PowerShell session, you'll have to import the module into your session again.

### 7.4.Create the WVD Tenant

1. In the PowerShell session, login as a Tenant Creator using the command below

```
Add-RdsAccount -DeploymentUrl https://rdbroker.wvd.microsoft.com
```



2. Now run the below commands to create a new WVD Tenant

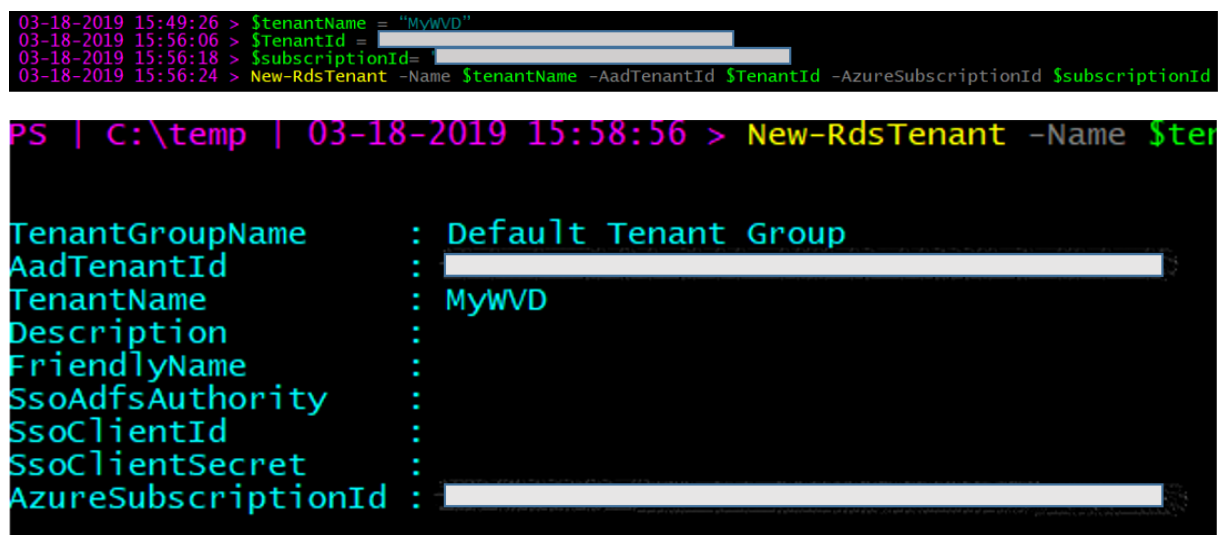
**#Setting Variables. Update the demo values within “ “ based on your specifics**

```
$tenantName = "MyWVD"
```

```
$TenantId = "00000000-0000-0000-000000000000"
```

```
$subscriptionId= "00000000-0000-0000-000000000000"
```

```
New-RdsTenant -Name $tenantName -AadTenantId $TenantId -  
AzureSubscriptionId $subscriptionId
```



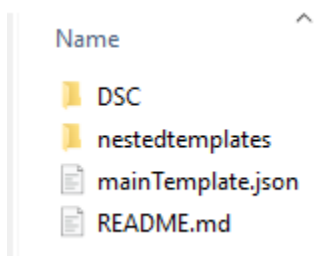


## 7.5.Create Host Pools using ARM template

Host pools are a collection of one or more identical virtual machines within Windows Virtual Desktop tenant environments. Each host pool can contain an app group that users can interact with as they would on a physical desktop.

Follow the steps in this article to create a host pool within a Windows Virtual Desktop tenant. This includes creating a host pool in Windows Virtual Desktop, creating a resource group with VMs in an Azure subscription, joining those VMs to the Active Directory domain, and registering the VMs with Windows Virtual Desktop.

1. Download the template from [here](#) and extract the files to a known path on your local system.



2. Now download the attached **parameters.json** file to the above folder



parameters.json

3. Choose **your preferred deployment style** from the below list and follow the instructions from that section to create your WVD host pools.
  - a. **Deploy VM using Custom Image:** Choose this option if you already have a custom VM image ready with your applications installed & configured. Follow, the step-step instructions in the section - [Deploy WVD Session Hosts using Custom Image](#) and continue from here once complete
  - b. **Deploy VM using custom VHD from blob storage:** Choose this option if you already have a custom VHD in an Azure storage Account. Follow, the step-step instructions in the section - [Deploy VM using custom VHD from blob storage](#) and continue from here once complete
4. Once all the parameters have been defined, the user can execute the deployment using PowerShell

**NOTE:** If the resource groups provided in the ARM template already exist, skip to the next step OR you are required to manually create the Resource Groups provided in the ARM template using the below commands.

```
#set AAD tenant ID (get this from Azure portal)
$tenantID = "00000000-0000-0000-000000000000"
```

```
# login to azure
Connect-AzureRmAccount -TenantId $tenantId
```

```
16:28:28 > $tenantID = 
16:30:35 > Connect-AzureRmAccount -TenantId $tenantId
```

```
#Create new RG and updates values within " " as
appropriately required.
New-AzureRmResourceGroup -Name "name" -Location "location"
```

```
16:30:35 > New-AzureRmResourceGroup -Name "MyRG" -Location "WestUS"
```

5. Now, Deploy the ARM template using PowerShell by running the following commands:

```
#set variables and update values appropriately
$RGName = "MyRG"
$TemplateFile = "C:\temp\folder\maintemplate.json"
$TemplateParameterFile = "C:\temp\folder\parameters.json"
```

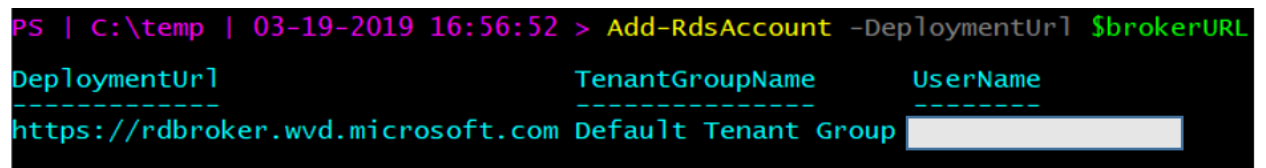
```
#create the hostpool
New-AzureRmResourceGroupDeployment -ResourceGroupName
$RGName -TemplateFile $TemplateFile -TemplateParameterFile
$TemplateParameterFile -Mode Incremental
```

```
16:37:40 > #set variables
16:37:43 > $RGName = "MyRG"
16:37:43 > $TemplateFile = "C:\temp\folder\maintemplate.json"
16:37:43 > $TemplateParameterFile = "C:\temp\folder\parameters.json"
16:37:43 > 
16:37:43 > #create the hostpool
16:37:43 > New-AzureRmResourceGroupDeployment -ResourceGroupName $RGName -TemplateFile $TemplateFile -TemplateParameterFile $TemplateParameterFile -Mode Incremental
```

6. Now, we will validate this newly created host pool
7. Open PowerShell and first connect to the WVD tenant using below commands.

```
#UPDATE THESE VALUES FIRST
$module = "C:\temp\RDPowerShell"
$TenantGroupName = "Default Tenant Group"

$brokerURL= "https://rdbroker.wvd.microsoft.com"
Import-Module $module\Microsoft.RDInfra.RDPowerShell.dll
Add-RdsAccount -DeploymentUrl $brokerURL
Set-RdsContext -TenantGroupName $TenantGroupName
```



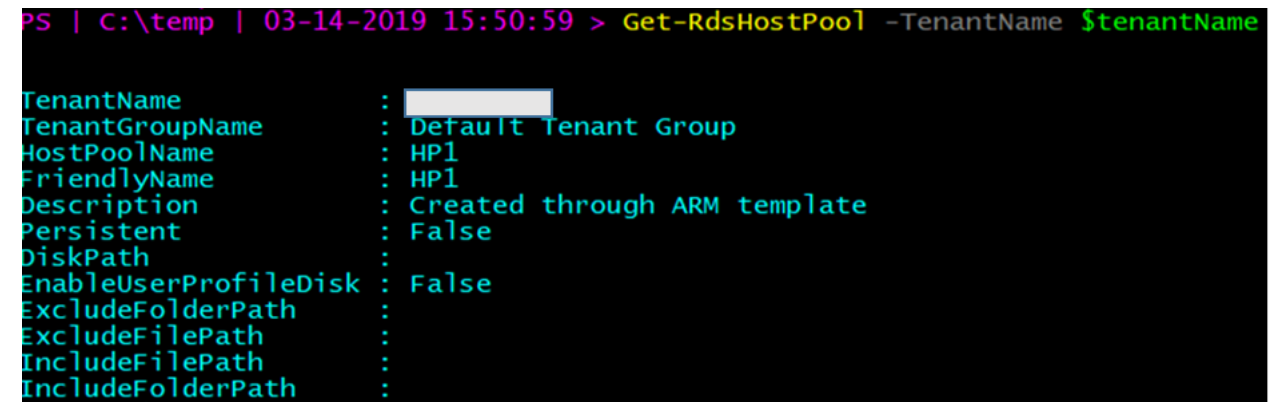
```
PS | C:\temp | 03-19-2019 16:56:52 > Add-RdsAccount -DeploymentUrl $brokerURL

DeploymentUrl          TenantGroupName        UserName
-----
https://rdbroker.wvd.microsoft.com Default Tenant Group
```

8. Check for the new Host Pool using below command

```
#UPDATE THESE VALUES FIRST
$TenantName = "MyWVD"

Get-RdsHostPool -TenantName $tenantName
```



```
PS | C:\temp | 03-14-2019 15:50:59 > Get-RdsHostPool -TenantName $tenantName

TenantName           : 
TenantGroupName      : Default Tenant Group
HostPoolName         : HP1
FriendlyName         : HP1
Description          : Created through ARM template
Persistent           : False
DiskPath             : 
EnableUserProfileDisk : False
ExcludeFolderPath    : 
ExcludeFilePath      : 
IncludeFilePath       : 
IncludeFolderPath     :
```

9. Check for the Session hosts in the Host Pool and ensure the status is **Available**

```
#UPDATE THESE VALUES FIRST
$HostPoolName = "HP1"
```

```
Get-RdsSessionHost -TenantName $tenantName -HostPoolName  
$HostPoolName
```

```
PS | C:\temp | 03-14-2019 15:56:40 > Get-RdsSessionHost -TenantName $tenantName -HostPoolName HP1  
  
SessionHostName : HP1-0  
TenantName      :  
TenantGroupName : Default Tenant Group  
HostPoolName    : HP1  
AllowNewSession : True  
Sessions        : 1  
LastHeartBeat   : 3/14/2019 10:57:43 PM  
AgentVersion    : 1.0.1.3  
AssignedUser    :  
Status          : Available  
StatusTimestamp : 3/14/2019 10:57:43 PM  
  
SessionHostName : HP1-1  
TenantName      :  
TenantGroupName : Default Tenant Group  
HostPoolName    : HP1  
AllowNewSession : True  
Sessions        : 1  
LastHeartBeat   : 3/14/2019 10:57:38 PM  
AgentVersion    : 1.0.1.3  
AssignedUser    :  
Status          : Available  
StatusTimestamp : 3/14/2019 10:57:38 PM
```

## 7.6.Manage App Groups

The default app group is automatically created for a new host pool that publishes the full desktop. In addition, you can create one or more application groups for the host pool. In this section, we will create a RemoteApp AppGroup and publish individual Start menu apps.

1. Check the Default Desktop Application Group is automatically created using below command

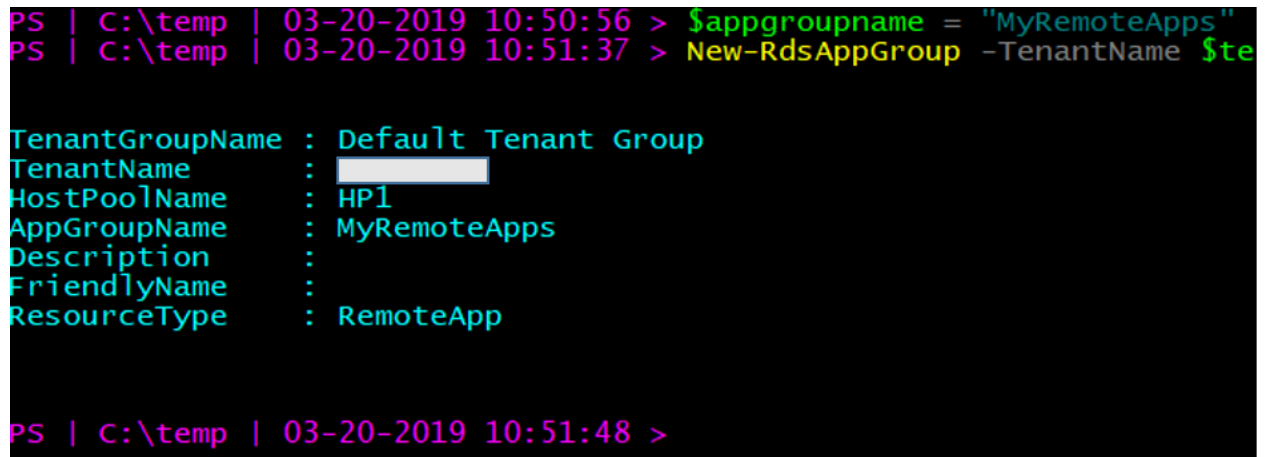
```
Get-RdsAppGroup -TenantName $tenantName -HostPoolName  
$HostPoolName
```

```
PS | C:\temp | 03-14-2019 15:57:46 > Get-RdsAppGroup -TenantName $tenantName -HostPoolName HP1  
  
TenantGroupName : Default Tenant Group  
TenantName      :  
HostPoolName    : HP1  
AppGroupName    : Desktop Application Group  
Description      : The default desktop application group for the session host pool  
FriendlyName    : Desktop Application Group  
ResourceType    : Desktop
```

- Now run the following PowerShell cmdlet to create a new empty RemoteApp group

```
#UPDATE THESE VALUES FIRST
$appgroupname = "MyRemoteApps"

New-RdsAppGroup -TenantName $tenantName -HostPoolName $hostpoolName -Name $appgroupname -ResourceType "RemoteApp"
```



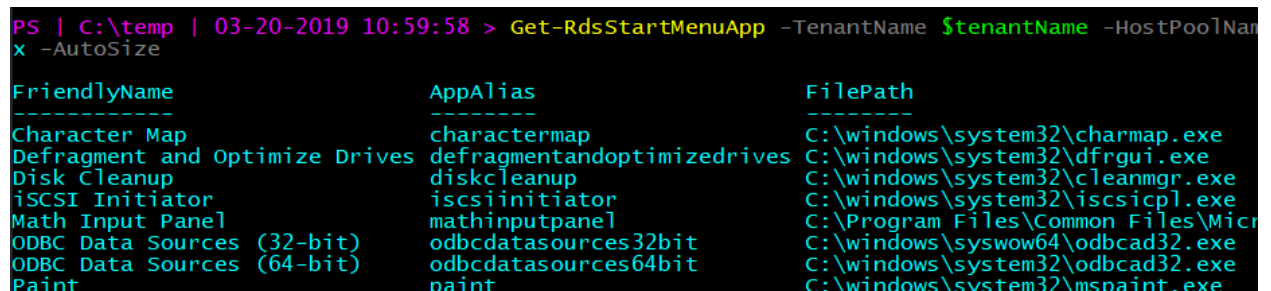
```
PS | C:\temp | 03-20-2019 10:50:56 > $appgroupname = "MyRemoteApps"
PS | C:\temp | 03-20-2019 10:51:37 > New-RdsAppGroup -TenantName $te

TenantGroupName : Default Tenant Group
TenantName      : 
HostPoolName    : HP1
AppGroupName     : MyRemoteApps
Description      : 
FriendlyName     : 
ResourceType    : RemoteApp

PS | C:\temp | 03-20-2019 10:51:48 >
```

- Run the following cmdlet to get a list of start menu apps on the host pool's virtual machine image. Write down the values for FilePath, IconPath, IconIndex, and other important information for the application you want to publish.

```
Get-RdsStartMenuApp -TenantName $tenantName -HostPoolName $hostpoolName -appgroupname $appgroupname | FT
FriendlyName,AppAlias,FilePath,IconPath,IconIndex -AutoSize
```



```
PS | C:\temp | 03-20-2019 10:59:58 > Get-RdsStartMenuApp -TenantName $tenantName -HostPoolName $hostpoolName -appgroupname $appgroupname | FT
x -AutoSize

FriendlyName      AppAlias      FilePath
-----
Character Map     charactermap   C:\windows\system32\charmap.exe
Defragment and Optimize Drives defragmentandoptimizedrives C:\windows\system32\dfrgui.exe
Disk Cleanup      diskcleanup    C:\windows\system32\cleanmgr.exe
iSCSI Initiator   iscsiinitiator C:\windows\system32\iscsicpl.exe
Math Input Panel  mathinputpanel C:\Program Files\Common Files\Micr
ODBC Data Sources (32-bit) odbcdatasources32bit C:\windows\syswow64\odbcad32.exe
ODBC Data Sources (64-bit) odbcdatasources64bit C:\windows\system32\odbcad32.exe
Paint             paint          C:\windows\system32\mspaint.exe
```

- Run the following cmdlet to publish a new RemoteApp to the application group and you will need the values from the above command to be used here.

#updates these variables with corresponding values form above command that you saved.

```
$name = "wordpad"
```

```
$filepath="C:\Program Files\Windows  
NT\Accessories\wordpad.exe"
```

```
$IconPath = "C:\Program Files\Windows  
NT\Accessories\wordpad.exe"
```

```
$IconIndex = 0
```

```
New-RdsRemoteApp -TenantName $tenantName -HostPoolName  
$hostpoolName -appgroupname $appgroupname -Name $name -Filepath  
$filepath -IconPath $IconPath -IconIndex $IconIndex
```

```
PS | C:\temp | 03-20-2019 11:13:46 > $appalias = "wordpad"
PS | C:\temp | 03-20-2019 11:14:16 > $filepath="C:\Program Files\Windows NT\Accessories\wordpad.exe"
PS | C:\temp | 03-20-2019 11:14:16 > $iconPath = "C:\Program Files\Windows NT\Accessories\wordpad.exe"
PS | C:\temp | 03-20-2019 11:14:16 > $iconIndex = 0
PS | C:\temp | 03-20-2019 11:14:16 > New-RdsRemoteApp -TenantName $tenantName -HostPoolName $hostpoolName  
-IconIndex $iconIndex

TenantGroupName : Default Tenant Group
TenantName      : 
HostPoolName    : HP1
AppGroupName     : MyRemoteApps
RemoteAppName   : wordpad
FilePath        : C:\Program Files\Windows NT\Accessories\wordpad.exe
AppAlias        :
```

Now, update the variables and repeat the above commands for any other applications you want to publish. As an example, we are publishing Paint & Snipping Tool in addition to WordPad.

5. To verify that the app was published, run the following cmdlet.

```
Get-RdsRemoteApp -TenantName $tenantName -HostPoolName  
$hostpoolName -AppGroupName $appgroupname | FT  
TenantName,HostPoolName,AppGroupName,RemoteAppName,ShowInWebFe  
ed,FilePath,IconPath,IconIndex
```

```
PS | C:\temp | 03-20-2019 11:20:25 > Get-RdsRemoteApp -TenantName $tenantName  
ShowInWebFeed,FilePath,IconPath,IconIndex

TenantName HostPoolName AppGroupName RemoteAppName ShowInWebFeed FilePath
-----
QLBL-WVD    HP1                MyRemoteApps paint          True C:\windows\s
QLBL-WVD    HP1                MyRemoteApps snippingtool  True C:\windows\s
QLBL-WVD    HP1                MyRemoteApps wordpad       True C:\Program F
```

6. Run the following cmdlet to grant users access to the RemoteApps in the app group

```
#UPDATE THESE VALUES FIRST
$appgroupname = "MyRemoteApps"
$upn = "rdsuser1@domain.com" #this should be the user that
will access WVD resources from your domain
```

```
Add-RdsAppGroupUser -TenantName $tenantName -HostPoolName
$HostPoolName -AppGroupName $appgroupname -UserPrincipalName
$upn
```

```
29:24 > $appgroupname = "MyRemoteApps"
29:46 > $upn = "rdsuser1@kloudeez.com"
29:54 > Add-RdsAppGroupUser -TenantName $tenantName -HostPoolName $HostPoolName -AppGroupName $appgroupname -UserPrincipalName $upn
30:14 >
```

```
#check the ACL has been applied using
```

```
Get-RdsAppGroupUser -TenantName $tenantName -HostPoolName
$HostPoolName -AppGroupName $appgroupname
```

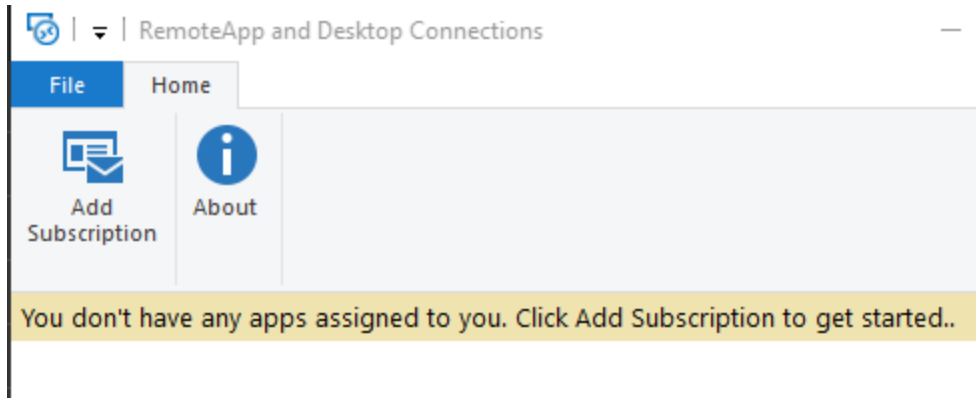
```
PS | C:\temp | 03-20-2019 11:30:14 > Get-RdsAppGroupUser -TenantName $tenantName -HostPoolName $HostPoolName -AppGroupName $appgroupname -UserPrincipalName $upn

UserPrincipalName : rdsuser1@kloudeez.com
TenantName       : 
TenantGroupName  : Default Tenant Group
HostPoolName     : HP1
AppGroupName     : MyRemoteApps
```

## 7.7. Validate user connections to WVD

At this stage, your RemoteApps are deployed on the WVD session hosts. A downloadable client is available that provides access to Windows Virtual Desktop resources from devices running Windows 7 and Windows 10 OR there is also a web client that can be used.

1. [Download the client](#) and run the MSI to complete the installation.
2. Start the client from the All Apps List, look for Remote Desktop.



3. Click Add Subscription > provide URL = <https://rdweb.wvd.microsoft.com/> > Next > Next Again



Enter your email address or connection URL

Email address or connection URL:

<https://rdweb.wvd.microsoft.com/>

Examples:

4. Sign in with you're the user account that was granted access to the WVD-RemoteApps in the earlier section > Next



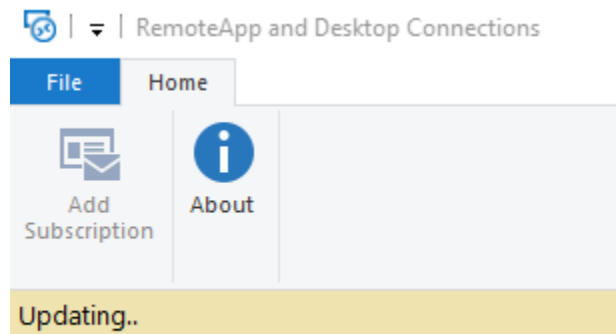
Let's get you signed in

Work or school, or personal Microsoft account

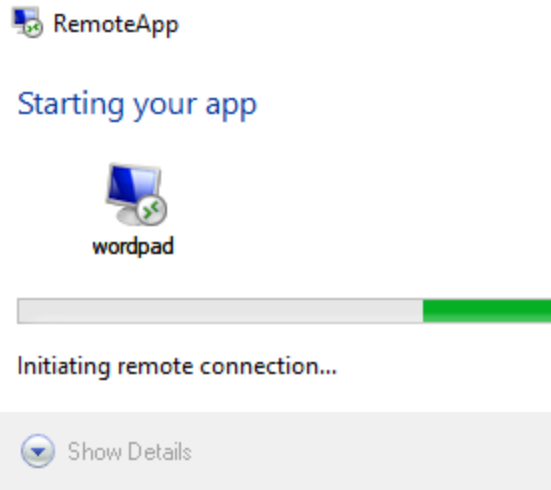
rdsuser1@kloudeez.com

Which account should I use?

Sign in with the username and password you use with Office 365 services from Microsoft.



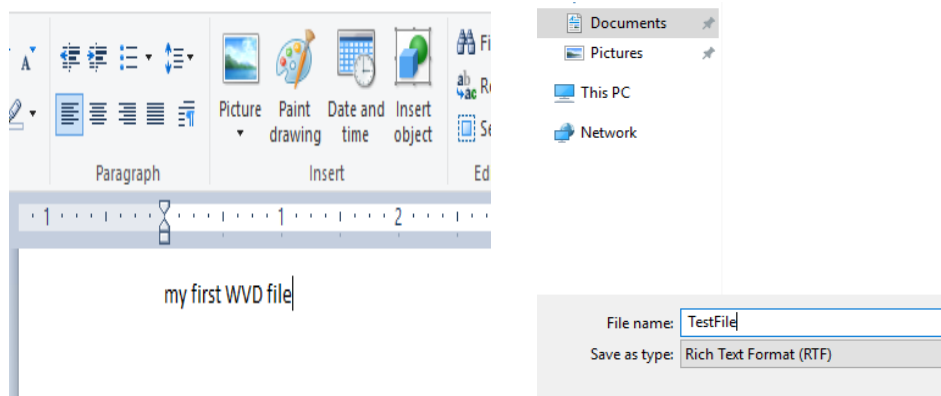
5. After successfully authenticating, you should now see a list of resources available to you.
6. Please launch any of the resources (EX: Wordpad). *please be advised that the first launch may be slow as your user profile is being created.*



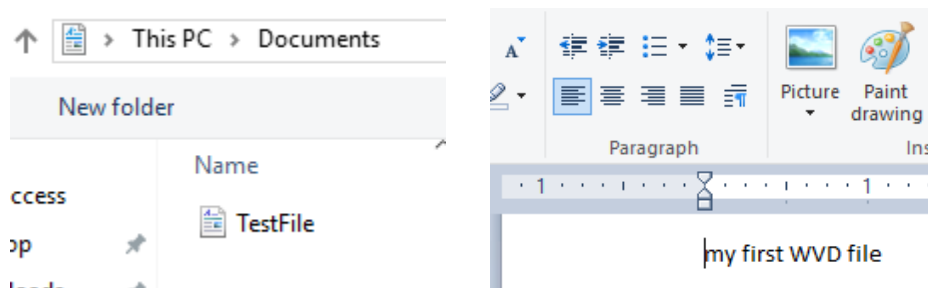
7. Once launched, you can see the icon in your taskbar



8. Now type something > save your file > close WordPad



9. Once again launch WordPad from the WVD client > Ctrl +O > to see you document present.



10. Alternatively, you can have a similar connection experience using a web browser by following the steps below.

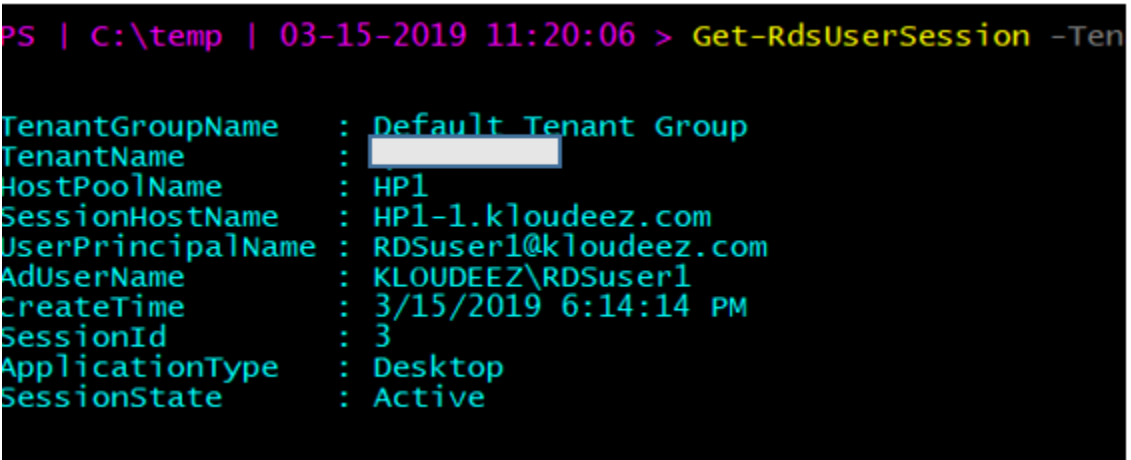
NOTE: the browser must be HTML-5 compatible. Supported ones include latest versions of IE/Edge/Safari/Firefox/Chrome

- Going to <https://rdweb.wvd.microsoft.com>
- Login with user domain credentials
- Access Apps & Desktops

11. As an Admin, you can also validate the User Session data from the WVD end using either of the commands.

```
#for all AppGroups in a HostPool
Get-RdsUserSession -TenantName $tenantName
```

```
#Filter to a specific for all AppGroup in a HostPool
Get-RdsUserSession -TenantName $tenantName -HostPoolName
$hostpoolName -Verbose
```



```
PS | C:\temp | 03-15-2019 11:20:06 > Get-RdsUserSession -Ten

TenantGroupName      : Default Tenant Group
TenantName            : 
HostPoolName         : HP1
SessionHostName      : HP1-1.kloudeez.com
UserPrincipalName     : RDSuser1@kloudeez.com
AdUserName            : KLOUDEEZ\RDSuser1
CreateTime           : 3/15/2019 6:14:14 PM
SessionId             : 3
ApplicationType       : Desktop
SessionState          : Active
```

## 8. Migrate on-prem VDI based RDS resources to Azure-WVD

Now that you have the experience of implementing and validating a successful WVD PoC, you can start migrating the on-premises VDI based RDS resources into Azure in a seamless fashion. The migration from a traditional VDI based RDS environment to WVD involves

some changes w.r.t the fact that the core server roles are not needed to be migrated and the focus would be on how to migrate the VDI Desktops to Azure.

## 8.1. Lift-n-Shift to Azure - Detailed Migration Steps

First party tools from Azure are available to Lift-n-Shift your On-premise infrastructure to Azure, namely Azure Site Recovery (ASR) and Azure Migrate (VMWare Migrations in Preview).

Based on the Hypervisor infrastructure used on-premises, the below table provides a reference point for the correct tool to be used for these operations

Infra	OS/Version	Assessment Tool	Migration Tool	WVD Connectivity
VMWare	Windows 7 Ent	AZ Migrate	AZ Migrate	Not Supported at the moment
VMWare	Windows 10 Ent	AZ Migrate	AZ Migrate	Supported
VMWare	Windows Server 2016	AZ Migrate	AZ Migrate	Supported
Hyper-V	Windows 7 Ent	AZ Migrate	ASR	Not Supported at the moment
Hyper-V	Windows 10 Ent	AZ Migrate	ASR	Supported
Hyper-V	Windows Server 2016	AZ Migrate	ASR	Supported

For example: Based on the above table, if you have client and/or server VM's operating on VMware, then Azure Migrate would be the correct fit Vs the VM's operating on HyperV have to use Azure Site Recovery (ASR). The next sections include guidance to help with the lift-n-shift operations using either scenario.

### 8.1.1. Hyper-V

1. Prepare Azure environment/resources for replicating On-premise VMs by following the guidelines [here](#).
  - i. Ensure that a new VNET, Resource Group isolated from your primary/production environment are created for the purposes of testing in ASR.
2. Once the Azure environment is setup as directed above, also prepare the on-premises Hyper-V by following the guidelines [here](#).
  - i. An ASR agent needs to be installed on the Hyper-V VM. Ensure you have appropriate permissions to perform the installation.
  - ii. Enable RDP on the VM to ensure connectivity after failover.

3. Depending on how you manage your VMs in Hyper-V, follow the guidelines below.
  - i. Managed by SCVMM – Follow guidelines [here](#) to setup VMs for Replication.
  - ii. NOT Managed by SCVMM - Follow guidelines [here](#) to setup VMs for Replication.
4. Perform a Test Failover of the replicated VMs in Azure to ensure all the data is being replicated properly and the VMs are functioning as they should. Please follow the guidelines [here](#).
  - i. Ensure you select a VNET, resource group that is separate from your primary/production environment.
  - ii. Since this VM is now in an isolated environment, it needs a Public IP to be able to accessible. Assign a public IP to the Test NIC and then RDP using this IP.
5. After a successful Test Failover, perform a Final Failover to Azure to successfully cutover the Hyper-V VM and start using the Azure VM by following the guidelines [here](#).
  - i. When performing a Final Failover, the VM needs to be in the primary/production VNET to ensure the VMs are talking to each other in your environment.

### 8.1.2. VMWare

Azure Migrate now support lift-and-shift migrations on VMWare environment. At the time of writing this document, the migration support is in **preview**. So before starting with the instructions in this section, you must first whitelist your subscriptions to use this service, using instructions at the link [here](#).

Users can continue using ASR to perform lift-and-shift migrations from VMWare environments. Please see the guidelines below.

1. Prepare Azure environment/resources for replicating On-premise VMs by following the guidelines [here](#).
  - i. Ensure that a new VNET, Resource Group isolated from your primary/production environment are created for the purposes of testing in ASR.

2. Once the Azure environment is setup as directed above, also prepare the on-premises VMWare by following the guidelines [here](#).
  - i. A VM needs to be imported when setting up replication. Ensure you have appropriate permissions to perform this step.
3. Setup Replication on your VMs by following the guidelines [here](#).
4. Perform a Test Failover of the replicated VMs in Azure to ensure all the data is being replicated properly and the VMs are functioning as they should. Please follow the guidelines [here](#).
  - i. Ensure you select a VNET, resource group that is separate from your primary/production environment.
  - ii. Since this VM is now in an isolated environment, it needs a Public IP to be able to accessible. Assign a public IP to the Test NIC and then RDP using this IP.
5. After a successful Test Failover, perform a Final Failover to Azure to successfully cutover the VMWare VM and start using the Azure VM by following the guidelines [here](#).
  - i. When performing a Final Failover, the VM needs to be in the primary/production VNET to ensure the servers are talking to each other in your environment.

Once all the VMs are replicated and fail-over into Azure is successful, Please follow the steps in this [section](#) to add the VM as a session host to a new or an existing hostpool and publish a Desktop app group and assign users to it.

## 9. Appendix

### 9.1 Deploy WVD Session Hosts using custom Image

**Deploy VM using Custom Image:** Choose this option if you already have a custom VM image ready with your applications installed & configured. Follow, the step-step instructions below

1. Goto the folder > open parameters.json *(using any editor that can update JSON files or even notepad)* > and update the following parameters
  - i. RdshImageSource = "CustomImage"
  - ii. RdshCustomImageSourceName = "My Azure VM Image" (change this to the actual Image name you want to use)

- iii. RdshCustomImageSourceResourceGroup = "My VM Image Resource Group" (change this to the actual Image name you want to use)
- iv. RdshUseManagedDisks = true
  - a. recommend to set to true and take advantage of managed disks.
  - b. IF setting to false, then update **StorageAccountResourceGroupName** = "ResourceGroupName" (this will be the Resource Group where the un-managed disks for your hostpool VM's will be stored)

```

"rdshImageSource": {
  "value": "CustomImage"
},
"rdshCustomImageSourceName": {
  "value": "My Azure VM Image"
},
"rdshCustomImageSourceResourceGroup": {
  "value": "My VM Image Resource Group"
},
"rdshUseManagedDisks": {
  "value": true
},
"storageAccountResourceGroupName": {
  "value": ""
},

```

- 2. Ensure that the below parameters are left empty
  - i. VmImageVhdUri
  - ii. RdshGalleryImageSKU
- 3. Update the VM configuration properties
  - i. RdshVmSize = "Standard\_D2S\_v3" (update this value based on your initial assessment recommendations. Refer this [link](#) for list of VM SKU)
  - ii. EnableAcceleratedNetworking = true OR false. (Default value is false. If setting to true, ensure that VM size selected above supports it. Typically, most of general purpose and compute-optimized instances with 2 or more vCPUs support this. On instances that supports hyperthreading it is required minimum of 4 vCPUs.)
  - iii. RdshNamePrefix (As an example, we are using HP1 as a VM prefix. So, all VM names in this Hostpool will be HP1-0, HP1-1, HP1-2 Etc.)

- iv. avSetPrefix (similar to above, enter the prefix of the Availability set name and the template will append "-availabilitySet" to it. For an already existing hostpool, if you want to add more session hosts, enter the part of the AV set name which precedes "-availabilitySet". For example, if the existing AV set name from the hostpool deployment is abc-availabilitySet you will enter abc as the avSetPrefix.
  - v. RdshNumberOfInstances (enter the total numbers of VMs [Ex: 2] you wish to deploy to this hostpool)
  - vi. RdshVMDiskType = SSD OR HDD (choose the appropriate disk type for the performance you seek)
4. Update the Domain & Network properties
- i. DomainToJoin = "Domain.com" (the domain FQDN to join the VMS to)
  - ii. ExistingDomainUPN = "admin@domain.com" (This UPN must have appropriate permissions to join the virtual machines to the domain and organizational unit)
  - iii. ExistingDomainPassword = "password" (the password to the account admin@domain.com)
  - iv. OUPath. = Defaults to "" (If you wish to add the VM to a specific path in AD, then update appropriately. EX: "OU=OrganizationalUnit,DC=Domain,DC=com")
  - v. ExistingVnetName = "Spoke-VNET" (or the VNET name created earlier)
  - vi. ExistingSubnetName = "default" (or the appropriate subnet created earlier)
  - vii. VirtualNetworkResourceGroupName = "VNET-RG-Name" (or the appropriate ResourceGroup Name where the VNET exists)
5. Update the WVD specific properties
- i. EnablePersistentDesktop = false (Default is False. Change to True to create the host pool for persistent desktops)
  - ii. DefaultDesktopUsers = "" (Leave as-is. We will assign users later)
  - iii. RdBrokerURL = "https://rdbroker.wvd.microsoft.com"
  - iv. ExistingTenantGroupName = "Default Tenant Group" (that's the default value but otherwise update if you have a different tenant group)
  - v. ExistingTenantName = "MyWVD" (Provide the WVD Tenant name created earlier)



- vi. HostPoolName = "RemoteAppsPool" (Provide a meaningful name based on what purpose the hostpool will serve. In this example the hostpool is a collection of RemoteApplications)
- vii. TenantAdminUpnorApplicationId= ["admin@domain.com"](mailto:admin@domain.com) OR "ApplicationIdGUID" (If you are creating a new host pool, this principal must be assigned either the RDS Owner or RDS Contributor role at the tenant scope (or higher). If you are registering these virtual machines to an existing host pool, this principal must be assigned either the RDS Owner or RDS Contributor role at the host pool scope (or higher))  
[!WARNING] You cannot enter a UPN that requires MFA to successfully authenticate. If you do, this template will create the virtual machines but fail to register them to a host pool.
- viii. TenantAdminPassword = "password for above account OR key for ServicePrincipal"
- ix. IsServicePrincipal = false (Default is False. Set to true only if you are providing an ApplicationId for TenantAdminUpnorApplicationId AND providing the respective ServicePrincipal Key for TenantAdminPassword)
- x. aadTenantId = "" (If you chose false for IsServicePrincipal above, then leave empty but if you chose true, then enter the AzureAD tenant ID)

## 9.2 Deploy WVD Session Hosts using custom VHD in blob storage

**Deploy VM using custom VHD from blob storage:** Choose this option if you already have a custom VHD in a Azure storage Account. Follow, the step-step instructions below

6. Goto the folder > open parameters.json *(using any editor that can update JSON files or even notepad)* > and update the following parameters
  - i. RdshImageSource = CustomVHD
  - ii. VmlImageVhdUri=["https://mystorageacc.blob.core.windows.net/vmimages/mysyspreppedVM.VHDX"](https://mystorageacc.blob.core.windows.net/vmimages/mysyspreppedVM.VHDX) (update to your VHDX path)
  - iii. RdshUseManagedDisks = true
    - a. recommend to set to true and take advantage of managed disks.
    - b. IF setting to false, then update  
**StorageAccountResourceGroupName** = "ResourceGroupName"  
(this will be the Resource Group where the un-managed disks for your hostpool VM's will be stored)

```

"rdshImageSource": {
  "value": "CustomImage"
},
"vmImageVhdUri": {
  "value": "https://mystorageacc.blob.core.windows.net/vmimages/mysyspreppedVM.VHDX"
},
"rdshUseManagedDisks": {
  "value": true
},
"storageAccountResourceGroupName": {
  "value": ""
},

```

7. Ensure that the below parameters are left empty
  - i. RdshGalleryImageSKU
  - ii. RdshCustomImageSourceName
  - iii. RdshCustomImageSourceResourceGroup

```

"rdshGalleryImageSKU": {
  "value": ""
},
"rdshCustomImageSourceName": {
  "value": ""
},
"rdshCustomImageSourceResourceGroup": {
  "value": ""
},

```

8. Update the VM configuration properties
  - i. RdshVmSize = "Standard\_D2S\_v3" (update this value based on your initial assessment recommendations. Refer this [link](#) for list of VM SKU)
  - ii. EnableAcceleratedNetworking = true OR false. (Default value is false. If setting to true, ensure that VM size selected above supports it. Typically, most of general purpose and compute-optimized instances with 2 or more vCPUs support this. On instances that supports hyperthreading it is required minimum of 4 vCPUs.)
  - iii. RdshNamePrefix (As an example, we are using HP1 as a VM prefix. So, all VM names in this Hostpool will be HP1-0, HP1-1, HP1-2 Etc.)
  - iv. avSetPrefix (similar to above, enter the prefix of the Availability set name and the template will append "-availabilitySet" to it. For an already existing hostpool, if you want to add more session hosts, enter the part of the AV set name which precedes "-availabilitySet". For example, if the existing AV

- set name from the hostpool deployment is abc-availabilitySet you will enter abc as the avSetPrefix.
- v. RdshNumberOfInstances (enter the total numbers of VMs [Ex: 2] you wish to deploy to this hostpool)
  - vi. RdshVMDiskType = SSD OR HDD (choose the appropriate disk type for the performance you seek)
9. Update the Domain & Network properties
- i. DomainToJoin = "Domain.com" (the domain FQDN to join the VMS to)
  - ii. ExistingDomainUPN = "admin@domain.com" (This UPN must have appropriate permissions to join the virtual machines to the domain and organizational unit)
  - iii. ExistingDomainPassword = "password" (the password to the account admin@domain.com)
  - iv. OUPath. = Defaults to "" (If you wish to add the VM to a specific path in AD, then update appropriately. EX: "OU=OrganizationalUnit,DC=Domain,DC=com")
  - v. ExistingVnetName = "Spoke-VNET" (or the VNET name created earlier)
  - vi. ExistingSubnetName = "default" (or the appropriate subnet created earlier)
  - vii. VirtualNetworkResourceGroupName = "VNET-RG-Name" (or the appropriate ResourceGroup Name where the VNET exists)
10. Update the WVD specific properties
- i. EnablePersistentDesktop = false (Default is False. Change to True to create the host pool for persistent desktops)
  - ii. DefaultDesktopUsers = "" (Leave as-is. We will assign users later)
  - iii. RdBrokerURL = "https://rdbroker.wvd.microsoft.com"
  - iv. ExistingTenantGroupName = "Default Tenant Group" (that's the default value but otherwise update if you have a different tenant group)
  - v. ExistingTenantName = "MyWVD" (Provide the WVD Tenant name created earlier)
  - vi. HostPoolName = "RemoteAppsPool" (Provide a meaningful name based on what purpose the hostpool will serve. In this example the hostpool is a collection of RemoteApplications)
  - vii. TenantAdminUpnOrApplicationId= ["admin@domain.com"](#) OR "ApplicationIdGUID" (If you are creating a new host pool, this principal must be assigned either the RDS Owner or RDS Contributor role at the

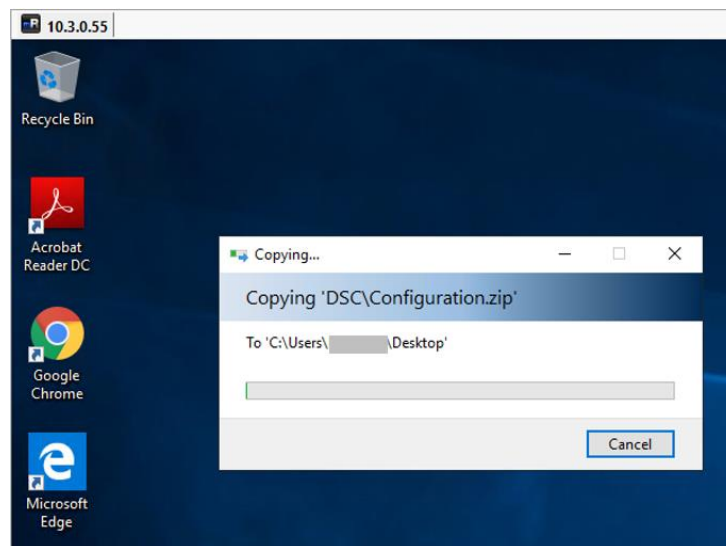
tenant scope (or higher). If you are registering these virtual machines to an existing host pool, this principal must be assigned either the RDS Owner or RDS Contributor role at the host pool scope (or higher))

[!WARNING] You cannot enter a UPN that requires MFA to successfully authenticate. If you do, this template will create the virtual machines but fail to register them to a host pool.

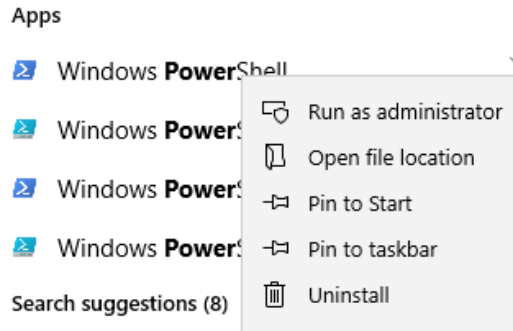
- viii. TenantAdminPassword = "password for above account OR key for ServicePrincipal"
- ix. IsServicePrincipal = false (Default is False. Set to true only if you are providing an ApplicationId for TenantAdminUpnorApplicationId AND providing the respective ServicePrincipal Key for TenantAdminPassword)
- x. aadTenantId = "" (If you chose false for IsServicePrincipal above, then leave empty but if you chose true, then enter the AzureAD tenant ID)

### 9.3 Install WVD Agents manually

1. Download the template and DSC bits from [here](#) onto the VM in Azure. Extract the zip file.



2. Run PowerShell as an Administrator.



3. CD into the DSC Folder you copied in the first step.

```
PS C:\Windows\system32> cd "C:\Users\██████████\Desktop\Create and provision WVD host pool\DSC"
PS C:\Users\██████████\Desktop\Create and provision WVD host pool\DSC>
```

4. Run the following command to install the AzureRM Module.
  - Install-Module -Name AzureRM -Force

```
PS C:\Users\██████████\Desktop\Create and provision WVD host pool\DSC> Install-Module -Name AzureRM -Force
Module provider is required to continue.
PowerShellGet requires Module provider version '2.0.0.0' or newer to interact with Module-based repositories. The Module provider must be available in 'C:\Program Files\PackageManagement\ProviderRepositories' or
'C:\Users\██████████\AppData\Local\PackageManagement\ProviderRepositories'. You can also install the Module provider by running 'Install-Module -Name PowerShellGet -MinimumVersion 2.0.0.0 -Force'. Do you want PowerShellGet to install and
import the Module provider now?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y
```

- This step might take a while to install all required modules and sub-modules.
5. Copy the following script into a Notepad and modify the fields as required and run in PowerShell to install the WVD Agents and either register a First Session Host to an existing empty HostPool or create a new one.

Note: To add any additional session hosts please refer point # 9 in this section.

```
$brokerURL = "https://rdbroker.wvd.microsoft.com"
$tenantName = "<<WVD Tenant Name>>"
$tenantGroup = "<<WVD Tenant Group>>"
$HostPoolName = "<<HostPool Name>>"
$TenantId = "<<Tenant ID>>"
$adJoinAdmin = "<<Admin UPN>>"
$ADAdminCredentials = New-Object
System.Management.Automation.PSCredential($adJoinAdmin,
(ConvertTo-SecureString "<<Admin Password>>" -AsPlainText
-Force))
$TenantAdminCredentials = New-Object
System.Management.Automation.PSCredential("<<Wvd tenant
admin UPN>>", (ConvertTo-SecureString "<<Admin
Password>>" -AsPlainText -Force))
```

```
Login-AzureRmAccount -TenantId $TenantId --Login with  
Global Admin Credentials
```

```
.\Script-FirstRdshServer.ps1 -RDBrokerURL $brokerURL -  
definedTenantGroupName $tenantGroup -TenantName  
$tenantName -HostPoolName $HostPoolName -Hours 24 -  
TenantAdminCredentials $TenantAdminCredentials -  
ADAdminCredentials $ADAdminCredentials -  
isServicePrincipal $true -AadTenantId $TenantId -  
EnablePersistentDesktop $false -Verbose
```

```
PS C:\Users\... Desktop\Create and provision vhd host pool\050> .\Script-FirstRdshServer.ps1 -RDBrokerURL $brokerURL -definedTenantGroupName $tenantGroup -TenantName $tenantName -HostPoolName $HostPoolName -Hours 24 -TenantAdminCredentials $TenantAdminCredentials -ADAdminCredentials $ADAdminCredentials -isServicePrincipal $true -AadTenantId $TenantId -EnablePersistentDesktop $false -Verbose
PS C:\Users\... Desktop\Create and provision vhd host pool\050> Get-RdsSessionHost -TenantName <tenant name> -wvd -HostPoolName <hostpool name>
SessionHostName : win10...
TenantName       : <tenant name> -wvd
TenantGroupName : Default Tenant Group
HostPoolName     : <hostpool name>
AllowNewSession  : True
Sessions         : 0
LastHeartBeat    :
AgentVersion     :
AssignedUser     :
Status           : Upgrading
StatusTimestamp  : 3/25/2019 10:14:09 PM
```

6. This will install all the required modules in the VM and if the Hostpool name specified in the parameters is new, a new hostpool will be created and the VM will be registered with it.

7. Check the status from PowerShell by running the following command against WVD. The status should say available.

- Get-RdsSessionHost -TenantName <<**tenant name**>> -  
HostPoolName <<**hostpool name**>>

```
PS C:\Users\...> Get-RdsSessionHost -TenantName <tenant name> -wvd -HostPoolName <hostpool name>
SessionHostName : win10...
TenantName       : <tenant name> -wvd
TenantGroupName : Default Tenant Group
HostPoolName     : <hostpool name>
AllowNewSession  : True
Sessions         : 0
LastHeartBeat    :
AgentVersion     :
AssignedUser     :
Status           : Upgrading
StatusTimestamp  : 3/25/2019 10:14:09 PM
```

- The status in the above image should change to Available for us to be able to use the Session host to publish applications/desktops.

```

PS C:\Users\Hemanth> Get-RdsSessionHost -TenantName [REDACTED] -wvd -HostPoolName [REDACTED]

SessionHostName : win10[REDACTED]
TenantName       : [REDACTED]-wvd
TenantGroupName  : Default Tenant Group
HostPoolName     : [REDACTED]
AllowNewSession  : True
Sessions         : 1
LastHeartBeat    : 3/25/2019 10:24:24 PM
AgentVersion     : 1.0.1.8
AssignedUser     :
Status           : Available
StatusTimestamp  : 3/25/2019 10:24:24 PM

```

8. Note: This process might take 10 to 15 minutes to complete.
9. After the first session host is registered, use the below script to add any additional session hosts to the same host pool.

```

$brokerURL = "https://rdbroker.wvd.microsoft.com"
$tenantName = "<<WVD Tenant Name>>"
$tenantGroup = "<<WVD Tenant Group>>"
$HostPoolName = "<<HostPool Name>>"
$TenantId = "<<Tenant ID>>"
$adJoinAdmin = "<<Admin UPN>>"
$ADAdminCredentials = New-Object
System.Management.Automation.PSCredential($adJoinAdmin,
(ConvertTo-SecureString "<<Admin Password>>" -AsPlainText
-Force))
$TenantAdminCredentials = New-Object
System.Management.Automation.PSCredential("<<Wvd tenant
admin UPN>>", (ConvertTo-SecureString "<<Admin
Password>>" -AsPlainText -Force))
Login-AzureRmAccount -TenantId $TenantId --Login with
Global Admin Credentials

.\Script-AdditionalRdshServers.ps1 -RDBrokerURL $brokerURL -
DefinedTenantGroupName $tenantGroup -TenantName $tenantName -
HostPoolName $HostPoolName -TenantAdminCredentials
$TenantAdminCredentials -ADAdminCredentials
$ADAdminCredentials -IsServicePrincipal $true -AadTenantId
$TenantId -Verbose

```

10. Once the session hosts are available, follow the guidelines [here](#) to publish applications/desktops as required.



## 9.4 Check Group Policy updates remotely

ONLY if you have PowerShell remoting enabled on your session hosts, with PowerShell using the below commands you can remotely update the VMs to get the latest group policy and also check the latest ones were applied

```
#update these values first  
$sessionhost = "HP1-0"
```

```
#update group policy  
ICM -ComputerName $sessionhost -ScriptBlock { gpupdate /force}
```

```
PS C:\Windows\system32> ICM -ComputerName HP1-0 -ScriptBlock { gpupdate /force}  
Updating policy...  
  
Computer Policy update has completed successfully.  
User Policy update has completed successfully.
```

```
#check if the new GPO was applied  
ICM -ComputerName $sessionhost -ScriptBlock { gpresult /r /scope  
computer}
```

```
PS C:\Windows\system32> ICM -ComputerName HP1-0 -ScriptBlock { gpresult /r /scope computer}  
Microsoft (R) Windows (R) Operating System Group Policy Result tool v2.0  
© 2018 Microsoft Corporation. All rights reserved.  
Created on 3/20/2019 at 9:44:20 PM  
  
RSOP data for on HP1-0 : Logging Mode  
-----  
OS Configuration:      Member Server  
OS Version:            10.0.17763  
Site Name:              Default-First-Site-Name  
Roaming Profile:  
Local Profile:  
Connected over a slow link?: No  
  
COMPUTER SETTINGS  
-----  
Last time Group Policy was applied: 3/20/2019 at 8:52:09 PM  
Group Policy was applied from: ADC01.kloudeez.com  
Group Policy slow link threshold: 500 kbps  
Domain Name: KLOUDEEZ  
Domain Type: Windows 2008 or later  
  
Applied Group Policy Objects  
-----  
RemoteDesktop-FSLogix-ProfileConfiguration  
RDS-Licensing  
Default Domain Policy  
  
The following GPOs were not applied because they were filtered out  
-----
```



## 10. Support

For support queries, please refer to our [support portal](#) where you can submit tickets to get additional assistance.