Azure Windows Virtual Desktop Network Reference Architecture

What is Windows Virtual Desktop

Azure Windows Virtual Desktop (WVD) is a desktop and application virtualization service that runs in the cloud. Customers are able to run a full desktop virtualization environment within their Azure subscription without the need to run any additional gateway servers as these are provided as a PaaS resource. A customer can provision and publish as many host pools as is required for individual workloads, bring custom images for production hosts, and even provide persistent desktops for individual users.

In addition to full desktops, customers can publish individual applications and assign specific users to custom app groups to reduce the total number of images. WVD removes the need for customers to manage Remote Desktop roles and instead use built-in delegated access to assign roles and collect diagnostics. As a managed service, WVD allows users to securely connect through reverse connections eliminating the need for inbound ports on the hosts or applications.

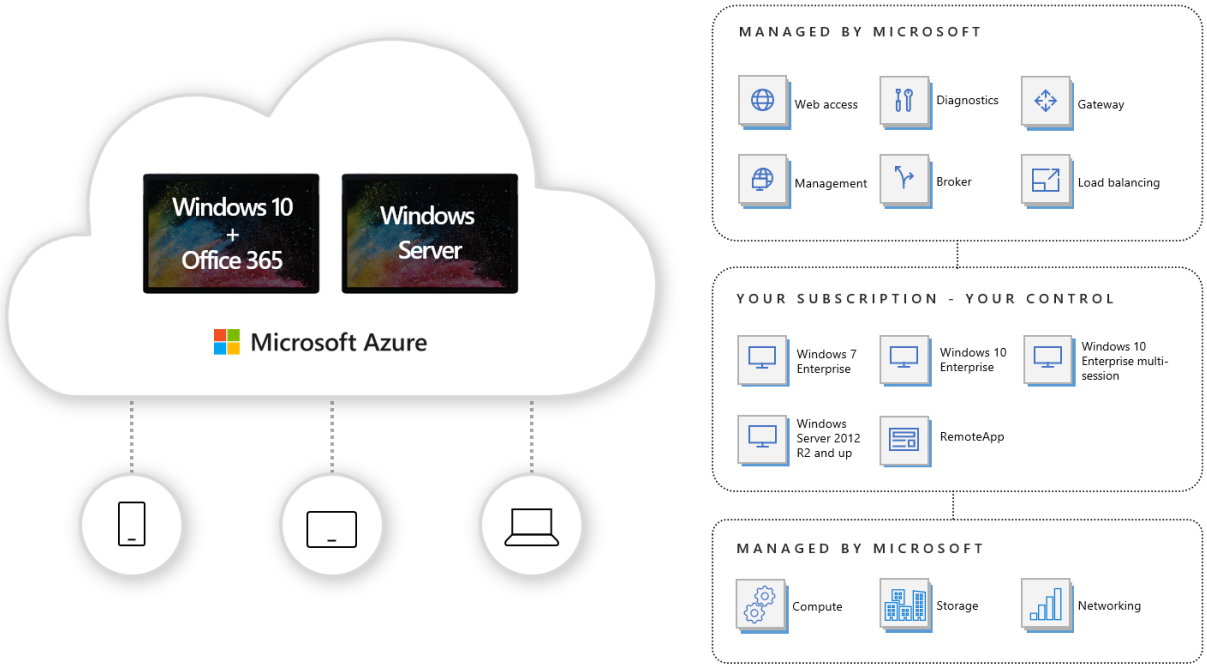


Figure 1: Windows Virtual Desktop Components

WVD Deployment Considerations

There are a few deployment scenarios which can impact some of the architecture options of the network connectivity. The most impactful consideration is how Active Directory is integrated into the environment. An Active Directory instance is required to which your backend desktops are able to join. The options for this are as follows:

Option	Pros	Cons
Use Azure AD DS.	Great for test or isolated environments that do not need connectivity to on-premises resources. Azure AD will be your leading source for identities.	AD DS will always be running, resulting in a fixed charge per month.
Spin up a DC in your Azure subscription.	Can sync with on-premises DCs if VPN or ExpressRoute is configured. All familiar AD Group Policies can be used. Virtual machines can be paused or stopped when needed to reduce costs.	Adds additional management of a VM and Active Directory in Azure.
Use VPN or ExpressRoute and make sure you're on-premises DCs can be found in Azure.	No AD DS or Domain Controller required in Azure.	Latency could be increased adding delays during user authentication to VMs. This assumes you have an on-premises environment, not suitable for cloud only tests.

The following architectures leverage the second option in which a domain controller is instantiated in Azure and Azure AD Connect was installed to sync identities to Azure AD.

IMPORTANT NOTE!

The applications being presented (independently or through desktops) via WVD may also impact the network requirements – specifically around network routing and ACLs. These requirements may have additional impacts on traffic flow and visibility.

WVD Network Architecture Options

While the WVD solution allows for multiple desktop and application offerings, this document will focus solely on the network architecture and the topologies for how this solution can be designed. This document will outline two different architectures: the first will describe a standard deployment in which WVD is deployed and users access desktops directly over the Internet, while the second will describe a high security deployment where access to desktops is privatized and all traffic is whitelisted and locked down for security auditing.

Standard WVD Deployment

For organizations looking to leverage WVD and allow users from the Internet to connect directly to desktops or applications, the high-level network architecture shown below is commonly deployed.

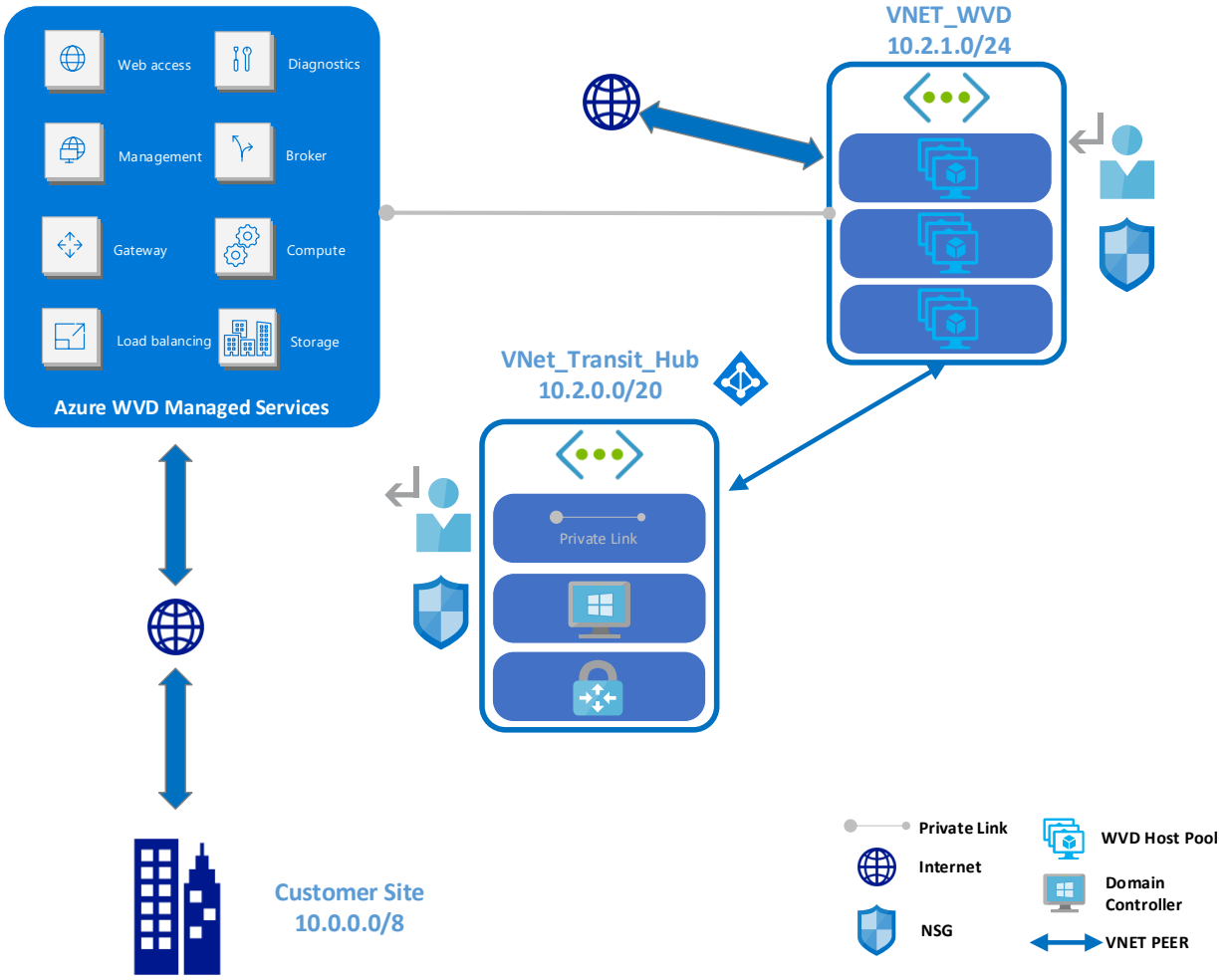


Figure 2: Internet Users Topology

In this model, external users will connect to the WVD gateway service which is accessible via the Internet. Azure Active Directory (AD) can be used to control access to the appropriate desktops and applications. Prior to any users connecting, the desktops in the backend pools proactively create an outbound connection to the Broker service. The Broker service and the Gateway service are part of the Microsoft managed PaaS services which work together to orchestrate a desktop/application session for users. As a result, there is no direct access into the backend desktop/applications via the Internet.

As a user attempts to connect to a desktop or remote application, the service ensures the user is authenticated to Azure AD, and the connection is established with the Gateway. The gateway service notifies the broker service of the incoming request and indicates an active connection to the user exists. The broker service determines the appropriate session host for the user, then instructs the appropriate backend resource to reach out to the gateway which has the active user connection. Once that

connection is complete the gateway server can facilitate the connection between the user and the backend desktop.

Advantages & Disadvantages

With any architecture there are advantages and disadvantages to the approach taken and it is important to understand how these affect your overall environment and service offering. For this approach some of the these include:

Advantages	Disadvantages
• Ease of configuration • Protection for backend hosts • Application level security for access	• Gateway accessible by malicious users • No visibility for outbound traffic • Open route tables on desktops

High-Security WVD Deployment

There are some organizations who have very strict security requirements and demand that all front-end access be private, like the on-prem user consumption previously detailed, as well as tightly lock down the back-end resources for least privilege access. In these scenarios all outbound communication must be tightly controlled via ACLs and any and all communication logged for visibility. These scenarios require all WVD dependencies to be clearly defined and whitelisted. A topology supporting this architecture is shown below.

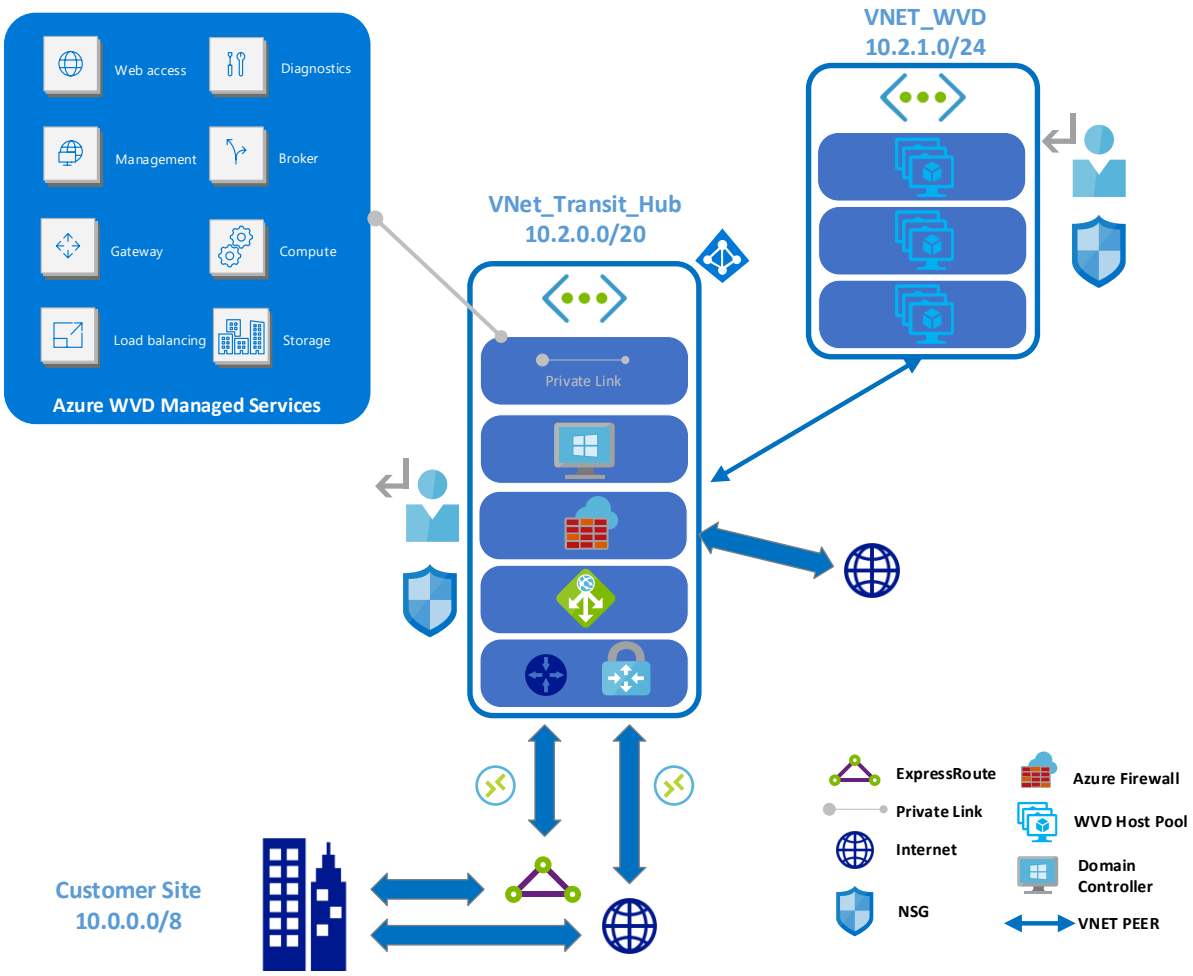


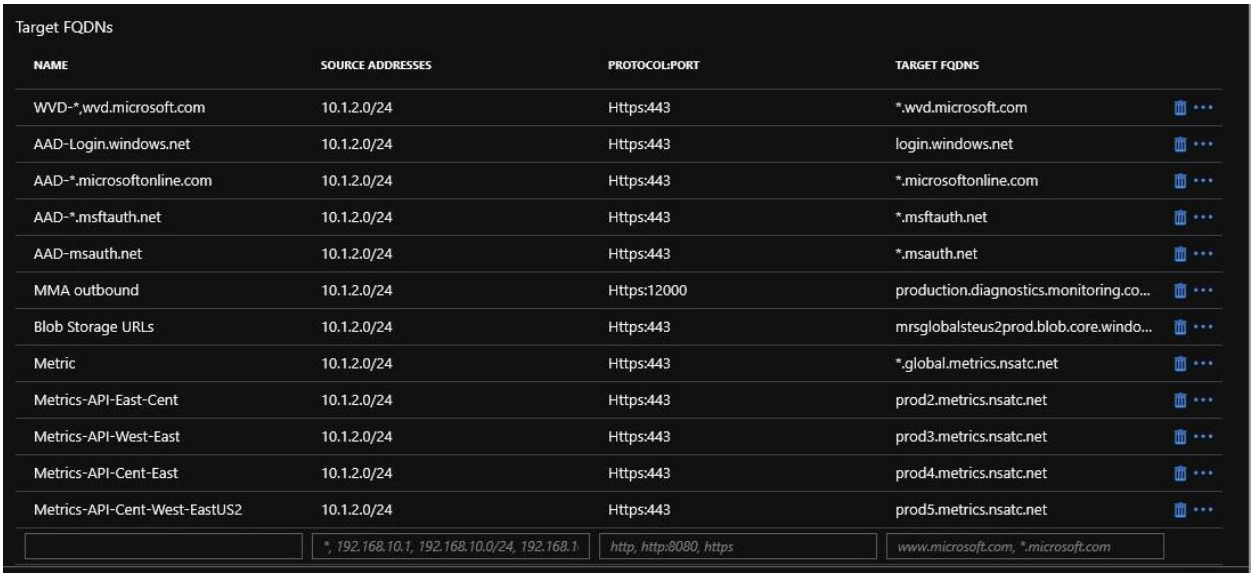
Figure 4: High-Security WVD Topology

This architecture builds upon the on-prem user consumption architecture as the front-end is fully privatized while also adding backend security to fully lock down and inspect all traffic destined to WVD dependencies. These dependencies are locked down via defined UDRs and Routes, service endpoints, as well as Azure Firewall (or NVA) and NSG logs for inspection. It is important to note that these dependencies only represent the WVD solution. Depending upon the applications being served to users through these desktops, additional dependencies may need to be accounted for.

Dependencies

- Blob Storage – The blob storage account is required to perform agent updates on the backend desktops. This can be handled by enabling service endpoints for Azure Storage on the backend desktop subnet however, a more secure way would be to leverage Azure Firewall and FQDN policies to whitelist and log the Blob Storage URL. Private Link will also be available for blob storage accounts.
- Geneva Monitoring URLs – Geneva is the Azure system used for internal monitoring. These URLs are defined per region and can be defined in an NVA or Azure Firewall. https://genevamondocs.azurewebsites.net/metrics/advanced/stamps.html

- Monitoring Agent – The monitoring agent on each desktop is used to make outbound calls to the monitoring service. This is an outbound call to production.diagnostics.monitoring.core.windows.net over port 12000. This can manually be defined in an NVA or Azure Firewall.
- WVD URLs – If private link is not used for the broker service or for all other WVD URLs these can be specified with *.wvd.microsoft.com in an NVA or using the App Service service tag built into Azure Firewall.
- Azure AD – For the backend desktops to reach out Azure AD, Azure Firewall can be leveraged and the default rule-sets will allow this traffic outbound as long as no explicit default deny rule is applied. The Azure firewall is a default deny rule so once these default rules are applied traffic will be dropped by default. These can also be defined manually in Azure Firewall or an NVA. https://docs.microsoft.com/en-us/azure/firewall/infrastructure-fqdns
- (OPTIONAL) Azure Key Vault – Key Vault may be used for credential management. If this is used this can be handled via private link as well.



Target FQDNs

NAME	SOURCE ADDRESSES	PROTOCOL:PORT	TARGET FQDNS
WVD-*.wvd.microsoft.com	10.1.2.0/24	Https:443	*.wvd.microsoft.com
AAD-Login.windows.net	10.1.2.0/24	Https:443	login.windows.net
AAD-*.microsoftonline.com	10.1.2.0/24	Https:443	*.microsoftonline.com
AAD-*.msftauth.net	10.1.2.0/24	Https:443	*.msftauth.net
AAD-msauth.net	10.1.2.0/24	Https:443	*.msauth.net
MMA outbound	10.1.2.0/24	Https:12000	production.diagnostics.monitoring.co...
Blob Storage URLs	10.1.2.0/24	Https:443	mrsglobalsteus2prod.blob.core.windo...
Metric	10.1.2.0/24	Https:443	*.global.metrics.nsatc.net
Metrics-API-East-Cent	10.1.2.0/24	Https:443	prod2.metrics.nsatc.net
Metrics-API-West-East	10.1.2.0/24	Https:443	prod3.metrics.nsatc.net
Metrics-API-Cent-East	10.1.2.0/24	Https:443	prod4.metrics.nsatc.net
Metrics-API-Cent-West-EastUS2	10.1.2.0/24	Https:443	prod5.metrics.nsatc.net
	*, 192.168.10.1, 192.168.10.0/24, 192.168.1	http, http:8080, https	www.microsoft.com, *.microsoft.com

Figure 5: Azure Firewall Rules

Routing

As we have all of the appropriate FQDNs defined in the upstream firewall (or with service endpoints or private link) we can apply a default route to the desktop subnet and route all traffic to the Azure Firewall. This will ensure that no traffic is destined outbound and we are able to gain visibility into all traffic leaving the environment.

As additional applications or integration with other Microsoft services are added (i.e. KeyVault, Netapp, etc.) it is important to understanding the routing needs to ensure routing symmetry and control.

Network Security Groups (NSG)

As we have a default route to the Azure Firewall, or NVA, security can be controlled via the firewall. If you would like to add additional NSGs to the desktops the list below represents the default outbound ports needed. No inbound connectivity is required so this can be left to deny all. Keep in mind that the ruleset defined below only represents the base rules needed for WVD to work. Additional rulesets may be required depending on other functions needed by the desktop.

- TCP/UDP 53 – DNS
- TCP 389 - LDAP
- TCP 445 - SMB
- TCP 443 – HTTPS

Private Link and DNS

There are 2 main FQDNs for the front-end access to WVD. These services can be deployed via Private Link in Azure so that an IP from the VNET is designated for these endpoints. On-prem DNS can then be pointed to these endpoints to route traffic privately over the hybrid connection or a DNS Proxy can be used to make these calls from within Azure https://github.com/microsoft/PL-DNS-Proxy.

- https://rdgateway.wvd.microsoft.com – this represents the gateway service the WVD users will connect to.
- https://rdweb.wvd.microsoft.com – this represents the feeds which WVD users will use in the remote desktop client to pull user specific resources.

When private link is deployed for the broker service, the URL https://rdbroker.wvd.microsoft.com is created and an A record within Azure DNS will be created to redirect desktops automatically to this private IP.

IMPORTANT NOTE!

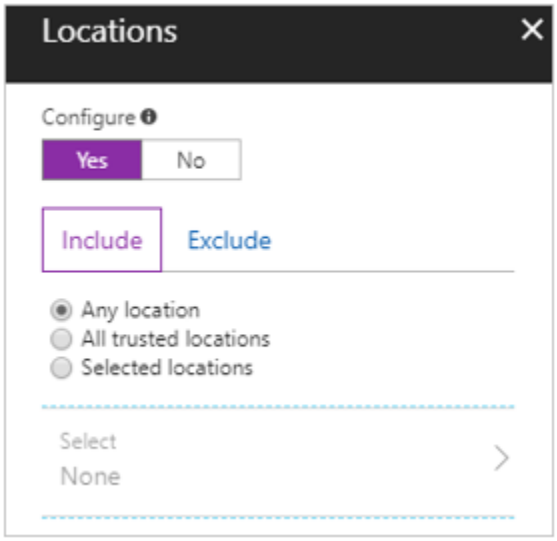
When using Azure Firewall rule processing order is important. Application rulesets are applied last so if Network rulesets are defined that match specified traffic, a corresponding application ruleset will never be hit. This is important to remember in use cases where desktops require Internet access as an Any 443 rule would supersede the FQDNs defined above. If Internet access is required for desktops, consider using a proxy service to prevent Any rules in the Azure Network rulesets.

Controlling Accessibility

The methods above have created a private path into the WVD environment, however accessibility from the Internet is still available. The front-end access for WVD (i.e. the broker and web services) are a shared platform with a shared DNS name across customers. Access is controlled via Active Directory as we have previously explained however we are not able to block Internet access to this front-end as this would impact all customers using the service. In order for customers to ensure that their environment is only accessible over private connectivity, we must leverage Azure Active Directory Conditional Access to limit the source of our requests to on-prem IP Addresses. https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/

A challenge today is that Azure Active Directory is not accessible via a private IP address therefore, we are not able to limit these requests to private IP addresses. We either must whitelist the public IP address for our enterprise organization or, if ExpressRoute is in use, we can enable Microsoft peering and receive the public IP addresses for Azure Active Directory over this private path. For Express Route Microsoft peering, we can then enable our NAT IP on the peering as the Trusted IP to allow authentication requests.

Using Conditional Access, we are able to define these Trusted IPs which represent an enterprises local network and limit access to our WVD environment to only requests from these IPs. https://docs.microsoft.com/en-us/azure/active-directory/authentication/howto-mfa-mfasettings#trusted-ips



Once we limit access to the WVD environment to only the Trusted locations defined for the Enterprise network, the WVD environment will not be accessible by users coming from the Internet. Desktops and Applications will only be accessible by those users on the corporate network with approved Active Directory permissions.

Advantages & Disadvantages

With any architecture there are advantages and disadvantages to the approach taken and it is important to understand how these affect your overall environment and service offering. For this approach some of the these include:

Advantages	Disadvantages
• Private front-end access by users • Prevent public access to resources • Visibility for all outbound traffic	• Most complex setup