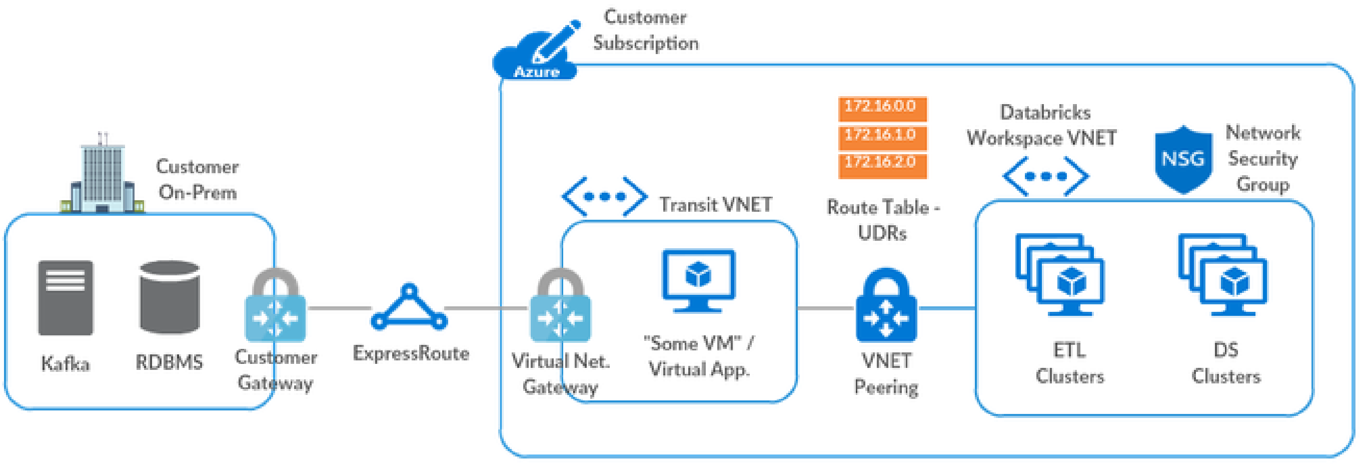
Azure Databricks On-Prem Connectivity Playbook

Step 0 - Overview

This is a high-level guide on how to establish connectivity from your Azure Databricks (ADB) Workspace to your on-premises network. It is based on the Hub-and-Spoke topology exemplified below, where traffic is routed via a transit VNET to the on-premises network.



Note: This process requires the VNET injection feature, where you're able to create a Azure Databricks workspace with a unmanaged VNET (basically BYOV - Bring Your Own VNET capability).

Pre-Conditions:

- Setup a VNET injected workspace.
- As shown above, on-premises connectivity requires presence of a Azure Virtual Network Gateway (ExpressRoute or VPN) in a transit VNET. Please skip to step 2 if one already exists.
- Please make sure to include the Microsoft and Databricks account teams. Do include them in all conversations related to this activity.

Step 1 - Setup Transit VNET with Virtual Network Gateway (If doesn't exist already)

If you already have ExpressRoute setup between on-premises and Azure, please follow this process to setup the virtual network gateway for a existing transit VNET. Else, please follow steps 1-5 here to create a transit VNET with a VPN-based virtual network gateway.

For any assistance during this part, please contact your local Microsoft account team.

Step 2 - Peer ADB VNET with Transit VNET

Please follow this process to peer the ADB VNET to transit VNET, but following options should be selected during the peering on both sides (as shown here):

- *Use Remote Gateways* on ADB VNET side of the peering
- *Allow Gateway Transit* on Transit VNET side of the peering

Note: Feel free to choose option *Allow Forwarded Traffic* as well on both sides of the peering, though it's recommended to be done "if" overall setup doesn't work as intended with just above settings.

Ref: Feel free to read this for more information on above options

Step 3 - Create Custom Route Table and associate with ADB Subnets

This step is required because once the ADB VNET is peered with the transit VNET (using the virtual network gateway), Azure auto-configures all routes via the latter. This could start breaking cluster setup within the ADB workspace, as there might be no properly configured return route from cluster nodes to ADB control plane.

- Create a route table as exemplified here
- Add custom routes / UDRs for the following (lookup table below, based on your ADB region) for *Next hop type* as "Internet", as exemplified here
    - Control Plane NAT VIP
    - Webapp
    - Metastore
    - Blob Storage (Artifacts)
    - Blob Storage (Logs)
    - DBFS Root Storage account
        - Go to the managed resource group for the workspace, and locate the storage account.
        - Find the **Blob Service Endpoint** for that storage account (it should be something like https://dbstoragela7xzqxl3bkfk.blob.core.windows.net/). Resolve this host to its IP address by using `nslookup` or an equivalent command.
- Associate the route table with ADB subnets (both public and private), as exemplified here (ignore steps 1-7, or see this)
    - Once custom routing table has been attached to ADB subnets, it's not necessary to edit the outbound security rules in the NSG. One could choose to change the outbound rule for "Internet / Any" to be more specific, but it really doesn't matter as the routes will control the actual egress. This will be relevant for packet filtering using firewall appliance too (Please see Appendix B).

Note: If IP-based route doesn't work during the tests (see Step 4 below) OR instead of creating UDRs for Blob Storage at this point, create Service Endpoint for *Microsoft.Storage*, such that all Blob Storage traffic goes via Azure backbone

Note: During route table creation, select *BGP route propagation* as "Enabled" (You may need to change it to "Disabled" if overall setup doesn't work as intended - try this as last resort only)

Note: If you're going to access other PaaS Azure Data Services from Databricks, like CosmosDB, SQL DW etc., you'll need to add UDRs for those services as well to the route table. Resolve the endpoint to its IP address using `nslookup` or an equivalent command.

Step 4 - Validate the Setup

Please do this to validate the setup:

- Try and create a cluster in the ADB workspace
    - This is a big test. If this fails, try and change a few configurations as mentioned in "Notes" above
    - If it still fails, check if route table has all required UDRs (including Service Endpoint rather than UDR for Blob Storage)
    - If nothing works, only option is to reach out to your Microsoft and Databricks account teams (if not included already)
- Once the cluster is created, try connecting to a "on-prem" VM by doing a simple ping from a notebook (with *%sh*)
    - If the ping is unsuccessful, verify the effective routes for cluster nodes to see if combination of custom and default routes are being applied correctly
    - Refer this to diagnose the issue

Ref: Feel free to read this for how Azure applies effective routes

Appendix

Appendix A - Lookup Table for UDRs / Blob Storage FQDNs

ADB Region	Service	Public IP / CIDR	Optional - FQDN for Blob Storage (If required for firewall whitelisting - see Appendix B)
East US 2	Control Plane NAT	23.101.152.95/32	-
	Webapp	40.70.58.221/32	-
	Metastore	52.177.185.181/32	-
	Blob Storage (Artifacts)	52.239.184.10/32	dbartifactsprodeastus2.blob.core.windows.net
	Blob Storage (Logs)	52.241.88.36/32	dblogprodwestus.blob.core.windows.net
North Central US	Control Plane NAT	23.101.152.95/32	-
	Webapp	40.70.58.221/32	-
	Metastore	23.96.178.199/32	-
	Blob Storage (Artifacts)	40.116.232.96/32	dbartifactsprodncus.blob.core.windows.net
	Blob Storage (Logs)	52.241.88.36/32	dblogprodwestus.blob.core.windows.net
	Control Plane NAT	23.101.152.95/32	-

East US	Webapp	40.70.58.221/32	-
	Metastore	40.121.158.30/32	-
	Blob Storage (Artifacts)	52.226.8.148/32	dbartifactsprodeastus.blob.core.windows.net
	Blob Storage (Logs)	52.241.88.36/32	dblogprodwestus.blob.core.windows.net
West US 2	Control Plane NAT	40.83.178.242/32	-
	Webapp	40.118.174.12/32	-
	Metastore	13.66.226.202/32	-
	Blob Storage (Artifacts)	13.77.184.64/32	dbartifactsprodwestus2.blob.core.windows.net
	Blob Storage (Logs)	52.241.88.36/32	dblogprodwestus.blob.core.windows.net
South Central US	Control Plane NAT	40.83.178.242/32	-
	Webapp	40.118.174.12/32	-
	Metastore	13.66.62.124/32	-
	Blob Storage (Artifacts)	52.239.159.196/32	dbartifactsprodscus.blob.core.windows.net
	Blob Storage (Logs)	52.241.88.36/32	dblogprodwestus.blob.core.windows.net
West US	Control Plane NAT	40.83.178.242/32	-
	Webapp	40.118.174.12/32	-
	Metastore	23.99.34.75/32	-
	Blob Storage (Artifacts)	52.238.56.180/32	dbartifactsprodwestus.blob.core.windows.net
	Blob Storage (Logs)	52.241.88.36/32	dblogprodwestus.blob.core.windows.net
West Europe	Control Plane NAT	23.100.0.135/32	-
	Webapp	52.232.19.246/32	-
	Metastore	191.237.232.75/32	-
	Blob Storage (Artifacts)	52.239.141.68/32	dbartifactsprodwesteu.blob.core.windows.net
	Blob Storage (Logs)	52.239.141.36/32	dblogprodwesteurope.blob.core.windows.net
North Europe	Control Plane NAT	23.100.0.135/32	-
	Webapp	52.232.19.246/32	-
	Metastore	40.113.93.91/32	-
	Blob Storage (Artifacts)	52.239.137.228/32	dbartifactsprodnortheu.blob.core.windows.net
	Blob Storage (Logs)	52.239.141.36/32	dblogprodwesteurope.blob.core.windows.net
South East Asia	Control Plane NAT	52.187.0.85/32	-
	Webapp	52.187.145.107/32	-
	Metastore	104.43.15.0/32	-
	Blob Storage (Artifacts)	52.239.197.36/32	dbartifactsprodseap.blob.core.windows.net
	Blob Storage (Logs)	191.238.64.192/32	dblogprodseasia.blob.core.windows.net
East Asia	Control Plane NAT	52.187.0.85/32	-
	Webapp	52.187.145.107/32	-
	Metastore	52.175.33.150/32	-
	Blob Storage (Artifacts)	52.175.112.16/32	dbartifactsprodeap.blob.core.windows.net
	Blob Storage (Logs)	191.238.64.192/32	dblogprodseasia.blob.core.windows.net
UK West	Control Plane NAT	51.140.203.27/32	-
	Webapp	51.140.204.4/32	-
	Metastore	51.141.8.11/32	-
	Blob Storage (Artifacts)	51.141.128.100/32	dbartifactsprodukwest.blob.core.windows.net

	Blob Storage (Logs)	51.141.128.100/32	dblogprodukwest.blob.core.windows.net
UK South	Control Plane NAT	51.140.203.27/32	-
	Webapp	51.140.204.4/32	-
	Metastore	51.140.184.11/32	-
	Blob Storage (Artifacts)	51.141.128.36/32	dbartifactsproduksouth.blob.core.windows.net
	Blob Storage (Logs)	51.141.128.100/32	dblogprodukwest.blob.core.windows.net
Australia East	Control Plane NAT	13.70.105.50/32	-
	Webapp	13.75.218.172/32	-
	Metastore	13.75.149.87/32	-
	Blob Storage (Artifacts)	52.239.131.36/32	dbartifactsprodauste.blob.core.windows.net
	Blob Storage (Logs)	52.239.131.36/32	dblogprodausteast.blob.core.windows.net
Australia Southeast	Control Plane NAT	13.70.105.50/32	-
	Webapp	13.75.218.172/32	-
	Metastore	191.239.192.109/32	-
	Blob Storage (Artifacts)	52.239.132.164/32	dbartifactsprodaustse.blob.core.windows.net
	Blob Storage (Logs)	52.239.131.36/32	dblogprodausteast.blob.core.windows.net

Appendix B - Route ADB traffic via Virtual Appliance / Firewall

You might also want to filter / vet all outgoing traffic from ADB cluster nodes using a firewall / DLP appliance (Azure Firewall, Palo Alto, Barracuda etc.). It may be required for something like:

- Satisfy enterprise security policy / controls which may mandate all outgoing traffic to be "inspected" and allowed / denied as configured, AND / OR
- Get a single NAT-like public IP / CIDR for all ADB clusters, which could be configured in ADLS firewall rules etc.
    - Single NAT IP for ADLS firewall rules hasn't been proven yet - It's recommended to use ADLS Service Endpoint instead (currently in preview as *Azure Active Directory*)

Process to do this remains almost the same as described above from Steps 0-4, but with few additional steps (this is just for reference, but details may vary per firewall appliance):

- Setup Virtual Appliance / Firewall within the transit VNET, as exemplified here
    - Alternative is to create the firewall in a *secure / DMZ* subnet within the Databricks VNET, which is separate from existing *private* and *public* subnets - Transit VNET solution is recommended as hub-spoke topology if firewall setup is required for multiple workspaces
- Create additional route in the custom route table (Step 3 above) to 0.0.0.0/0 with *Next hop type* as "Virtual Appliance" and appropriate *Ne xt hop address*
    - Routes configured in Step 3 should remain - though ones for Blob Storage could be removed (or remove the service endpoint for Blob Storage) if all Blob Storage traffic needs to be routed via the firewall
    - If using the *secure / DMZ* subnet approach, you may need to create an additional route table to be associated with that subnet only. That should have a route to 0.0.0.0 with *Next hop type* as "Internet" or "Virtual network gateway", depending on if traffic is destined for public network directly or via an on-prem network.
- Configure allow / deny rules in the firewall appliance (different for each product)
    - If you've removed the routes for Blob Storage (as indicated in above point), those should be whitelisted in the firewall (refer the FQDNs in lookup table above)
    - You may also need to whitelist certain public repositories like for Ubuntu etc., to ensure the clusters are created alright

Appendix C - Configure Custom DNS

You might also wish to use your own DNS for name resolution. That is possible with VNET injected workspace. Please see the following for more information on how one could configure custom DNS for a Azure VNET:

- Name resolution that uses your own DNS server
- Specify DNS servers