Azure Private Link Preview Instructions

(Only for Approved Preview Customers)

Pre-Requisites

- Please make sure that your Azure subscription has been enabled by Microsoft and Databricks
- For the User to Workspace (Frontend) Private Endpoints, please make sure that you've private connectivity from your on-prem network to your Azure network using Expressroute or VPN
- Please make sure that the user performing the setup has necessary read/write permissions to provision the Azure Databricks workspace and to provision the Private Endpoint in the transit / bastion & workspace VNETs

Summary

This is a short-version of what's needed to create a Private Link-enabled workspace (step 6 can be combined with 2 as 3-5 are specific to the User-to-Workspace use case):

1. [Automated] Provision the Azure Databricks workspace with *publicNetworkAccess*: *Disabled*, *requiredNsgRules*: *AllRules* or *NoAzureDatabricksRules*, and *enableNoPublicIp*: *true*. These settings **can not** be updated for a workspace today, so please configure carefully.
2. [Automated] Provision Private Endpoints for User-to-Workspace (in transit/bastion VNET) and / or Data Plane-to-Control Plane Relay/Webapp (ideally in a separate subnet in the workspace VNET)
3. [Automated] Add CNAME Record for the OAuth flow for the User-to-Workspace private endpoint (to the private DNS zone)
4. [Automated] Ensure outbound traffic for the OAuth flow for the User-to-Workspace private endpoint (add NSG rules for *AzureActiveDirectory* and *AzureFrontDoor.Frontend*)
5. [Automated] Ensure outbound traffic for Azure Active Directory from the workspace subnets (add NSG rule for *AzureActiveDirectory*)
6. [Semi-Automated] Setup Internal DNS to map to User-to-Workspace private endpoint - automation for this would depend on your internal processes
7. [Automated] Provision Private Endpoint for the DBFS Storage Account (ideally in the same subnet as the Data Plane-to-Control Plane Relay/Webapp one)

Step 1 - Provision the Azure Databricks Workspace

Deploy an Azure Databricks Workspace in your own managed VNET with No-Public-IP (Secure Cluster Connectivity) and Private Link support (only possible with ARM Template for now):

- Either create the NSG and the VNET as part of the all-in-one deployment (ref ARM Template), or
- Reuse an existing VNET and:
    - Create a couple of dedicated subnets for the workspace, and delegate both to *Microsoft.Databricks/workspaces*
    - Create a empty NSG, and attach it to both workspace subnets - the rules will be injected by Azure Databricks
    - Deploy the workspace with VNET and Subnet references (ref ARM Template)

Please make sure that you provide the appropriate values for these properties / parameters in the ARM template (one **can not** update these values for a workspace today):

- Property *publicNetworkAccess* - value: *Disabled* or *Enabled*
- Property *requiredNsgRules* - value: *AllRules* or *NoAzureDatabricksRules*
- Parameter *enableNoPublicIp* - value: *true* - to enable secure cluster connectivity clusters.

Note: Please select the pricing tier for the workspace as "**premium**".

publicNetworkAccess	requiredNsgRules	Private Endpoints
Disabled	NoAzureDatabricksRules	• User to Workspace (Frontend) - Required • Data Plane to Control Plane (Backend) - Required • DBFS - Optional
Disabled	AllRules	Combination Not Allowed
Enabled	NoAzureDatabricksRules	• User to Workspace (Frontend) - Optional • Data Plane to Control Plane (Backend) - Required • DBFS - Optional

Enabled	AllRules	• User to Workspace (Frontend) - Optional • Data Plane to Control Plane (Backend) - Not Allowed • DBFS - Optional

Step 2 - Provision Private Endpoints to Azure Databricks Workspace and Secure Cluster Connectivity Relay

Now that the Azure Databricks workspace is ready, it's time to create either or both of the following private endpoints based on values of *publicNetworkAccess* and *requiredNsgRules* (refer table in Step 1):

- User to Workspace (Frontend) - to ensure that all traffic from user/client machines to the workspace UI, REST API, JDBC/ODBC coordinates could go over the customer private network and Azure network backbone
- Data Plane to Control Plane (Backend) - to ensure that all traffic from the clusters in data plane to the secure cluster connectivity relay and REST API could go over the customer private network and Azure network backbone

Ref ARM Template for steps below. One could also automate it using such an Azure Powershell Cmdlet. The same *privateLinkServiceId* and *groupIds* fields would be applicable for both private endpoints (User to Workspace / Frontend and Data Plane to Control Plane / Backend). Please provide the full resource Id for the workspace in the *privateLinkServiceId* field.

Note: It's possible to use the same single private endpoint for both use cases, as long as it's reachable from the user network (like on-prem) and the workspace subnets. But please consider any performance implications of that architecture since all frontend and backend traffic would flow over the same private endpoint and would be subjected to its throughput / bandwidth limits. Hence, we recommend the following:

- In case of User to Workspace (Frontend) private endpoint, the private endpoint VNET is recommended to be different from the workspace VNET, and it should ideally be closer to the source of your traffic. E.g. in case of traffic coming from on-prem, the private endpoint should be set up close to the ExpressRoute / VPN Gateway (like in a transit / bastion VNET).
- In the case of Data Plane to Control Plane (Backend) private endpoint, the private endpoint subnet could be in the workspace VNET (though separate from workspace

subnets) or in a peered VNET. Please make sure that traffic is allowed to ports 443 (for Azure Databricks Webapp) and 6666 (for Secure Cluster Connectivity Relay).

Note: Please make sure that the private endpoint subnets for both User to Workspace (Frontend) and Data Plane to Control Plane (Backend) use cases have network policies disabled.

Just like the ARM template, below manual steps apply to both private endpoints (User to Workspace / Frontend and Data Plane to Control Plane / Backend):

- In Azure Console, go to Azure Private Link Center -> Private Endpoints
- Press **+Add** to start the process for creating a private endpoint
- In **Basics**:
    - In **Project Details**, choose the appropriate Subscription & Resource group to provision the private endpoint
    - Provide a valid Name (should be a unique name across all workspace private endpoints) & select the Region (same as Azure Databricks workspace) under **Instance details**
    - Press *Next: Resource*
- In **Resource**:
    - Leave the default for Connection method.
    - Select the Subscription for the Azure Databricks workspace
    - Select the Resource type as *Microsoft.Databricks/workspaces*
    - Select the Resource as the above provisioned provisioned workspace
    - Select the Target sub-resource as *databricks_ui_api*
    - Press *Next: Configuration*
- In **Configuration**:
    - Choose the appropriate Virtual network and Subnet under **Networking** to host the private endpoint for the workspace (refer the note for which subnet to use for User to Workspace (Frontend) and Data Plane to Control Plane (Backend) private endpoints)
    - In **Private DNS Integration**, leave the default *Yes* for Integrate with private DNS zone, and select *(New) privatelink.azuredatabricks.net* in Private DNS zones
        - If you're creating both User to Workspace (Frontend) and Data Plane to Control Plane (Backend) private endpoints, please make sure that both endpoints are set up in different resource groups and map to different private DNS zones i.e. do not use the same DNS zone for both.
- Add optional **Tags**
- Press *Review + Create* and then *Create*

Note: If the private endpoints are created by a user without CONTRIBUTOR or OWNER permissions on the workspace resource, then somebody with those relevant permissions will need to approve the private endpoint connections.

Note: In the case of Data Plane to Control Plane (Backend) use case, please make sure that the workspace VNET is linked to the relevant Private DNS Zone mapped to the private endpoint.

Note: If you've enabled custom routing (optinall to a egress firewall) for your workspace subnets, then please make sure that traffic is allowed to DBFS storage account (if you're not creating a private endpoint for it as indicated in Step 7) and regional Artifact Storage Account, Log Storage Account, Metastore and Telemetry Event Hub. For metastore, you could alternatively use External Hive Metastore in your network.

At this time, the relevant private endpoints and corresponding private DNS zones should have been created.

Step 3 - Add CNAME Record for the OAuth flow for the User to Workspace (Frontend) Private Endpoint

Go to the private DNS zone corresponding to the User to Workspace (Frontend) private endpoint created in the last step above, and add an additional record so that the AAD OAuth flow for the Azure Databricks workspace works seamlessly:

- Press **+Record** set, which should open a **Add Record Set** window
- Enter Name as *eastus2-c2.pl-auth* (or see below for your region)
    - For your region, do a nslookup on the workspace URL, like `nslookup adb-1111111111111111.15.azuredatabricks.net`, which would return the regional CNAME like `eastus2-c2.azuredatabricks.net`. The value for the Name property should be *<domain prefix before azuredatabricks.net>.pl-auth*.
- Select Type as *CNAME*
- Keep TTL as *1* and TTL unit as *Minutes*
- Enter the workspace URL to the Alias, like *adb-{workspaceId}.{seq}.azuredatabricks.net* (should already have an A-record in the DNS zone)
- Press *OK*

This could be automated using such an ARM Template object or this Azure Powershell Cmdlet.

Step 4 - Ensure Outbound Traffic for the OAuth flow for the User to Workspace (Frontend) Private Endpoint

For the User to Workspace (Frontend) private endpoint, please ensure that the outbound NSG rules for the source network that'll be used to access the Azure Databricks workspace, allow outbound traffic for AAD OAuth flow. In this case, the source network refers to wherever your workspace request traffic is coming from towards the configured private endpoint. E.g., it would be the ExpressRoute / VPN Gateway network (like a transit / bastion VNET) in case of requests coming from on-prem.

- *AzureActiveDirectory* (Port 443, TCP) - Used for the OAuth flow with AAD
    - There's a private preview for AAD Private Link too. So as an alternative to the NSG rule, a private endpoint could be created for the AAD traffic, which would remove the need for the above NSG rule.
    - Above mentioned private preview should work if your subscriptions are enabled for Azure Databricks Private Link, but please contact the AAD team in case you face any issues and if there's a specific enablement needed.
    - An alternative to NSG rule and private endpoint is the subnet-level service endpoint for *Microsoft.AzureActiveDirectory*.
- *AzureFrontDoor.Frontend* (Port 443, TCP) - Used to serve static assets from an Azure CDN for the AAD OAuth flow
- Any other IP addresses for AAD integrated SSO (in case AAD is federated with another IdP)

Everything else could be denied from an outbound access standpoint, since rest should just use the private endpoint for the workspace. Also, please note this NSG limitation for Private Endpoints.

This could be automated using this Azure Powershell Cmdlet.

Note: If you're using the same source network for accessing multiple workspaces through their own resource-level private endpoints, this step needs to be done only once across those workspaces. You need to do this again only if you're using a different source network than what's already used to access a workspace(s). E.g. if you access a workspace through ExpressRoute / VPN Gateway (from on-prem through transit / bastion VNET), and also from VMs / Apps in a Azure VNET, this step will need to be repeated for both those source networks.

Step 5 - Ensure Outbound Traffic for Azure Active Directory from the workspace subnets

Some of the product features like Mount Points, Credential Passthrough etc. require access to Azure Active Directory from the clusters in the workspace i.e. from the workspace's subnets. If you're configuring the private endpoint for Data Plane to Control Plane (Backend) and will restrict all outbound traffic other than what's required by Databricks, please add an outbound rule for *AzureActiveDirectory* (Port 443, TCP) to your workspace NSG.

As an alternative to the extra NSG rule, you could also create a subnet service endpoint for *Microsoft.AzureActiveDirectory*, or create a private endpoint for Azure Active Directory (it's in preview).

Step 6 - Setup Internal DNS to map to User to Workspace (Frontend) Private Endpoint

In context of the User to Workspace (Frontend) private endpoint, if you'll access the Azure Databricks workspace from on-prem network or an Azure VNET where you've enabled custom DNS, please make sure that appropriate setup is done so that the private endpoint IP for the workspace is accessible using the workspace URL.

You may need to configure relevant conditional forwarding for **.azuredatabricks.net* or **.privatelink.azuredatabricks.net* to Azure-provided DNS, or create A-record for the workspace URL in your on-prem / internal DNS, or perform other steps similar to what you would do to enable access to other Private Link enabled services. Please refer to Private Endpoint DNS.

E.g. some of our customers directly map the resource URLs to the User to Workspace (Frontend) private endpoint IPs in their internal DNS, which needs two entries:

- Map per-workspace URL like *adb-1111111111111.15.azuredatabricks.net* to the User to Workspace (Frontend) private endpoint IP, and
- Map AAD OAuth flow Reply URL like *eastus2-c2.pl-auth.azuredatabricks.net* (ref Step 3 above) to the User to Workspace (Frontend) private endpoint IP - this would be needed only once across workspaces as there's no workspace-level info shared on that. Though you could also configure this traffic to egress via the public network if sharing a single User to Workspace (Frontend) private endpoint IP for all teams / businesses would be a problem due to common DNS.

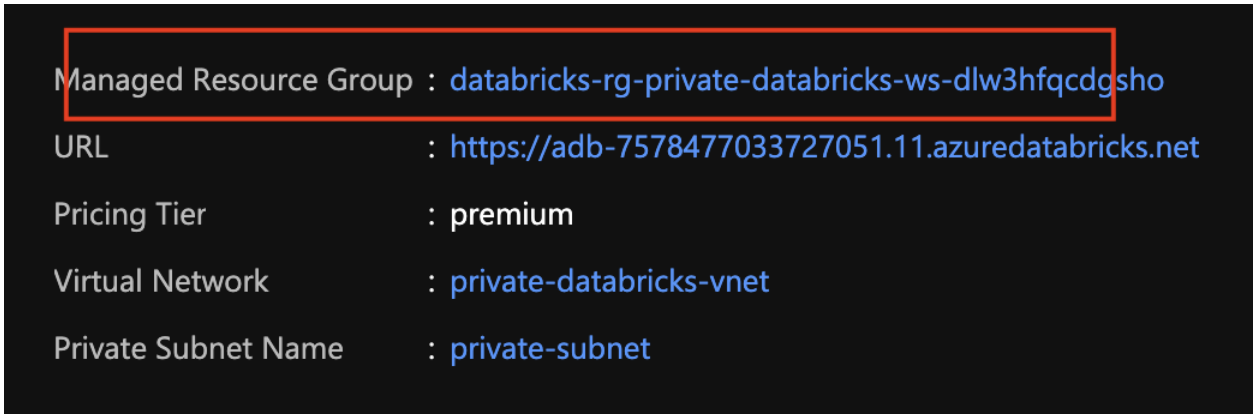
Once this is all set, you should be able to access the Azure Databricks workspace and start clusters for your workloads.

Note: The Data Plane to Control Plane (Backend) private endpoint also uses the same per-workspace URL DNS scheme i.e. *adb-1111111111111.15.azuredatabricks.net*. Under the hood, it's accessed via two ports - 443 for the Webapp and 6666 for the Secure Cluster Connectivity Relay.

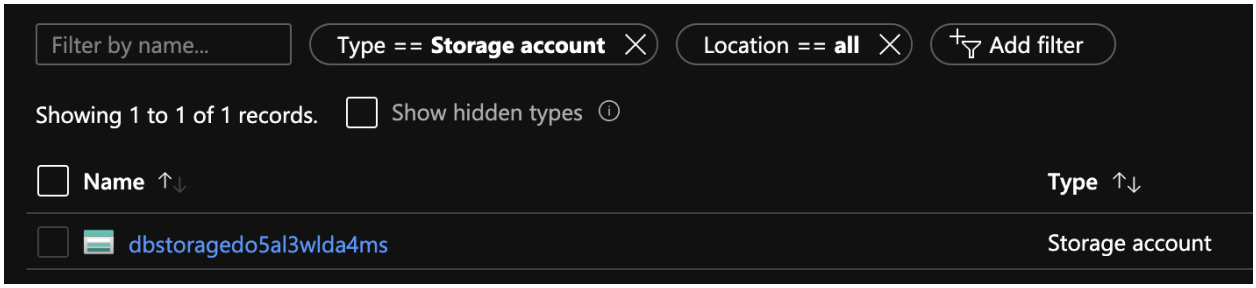
Step 7 - Provision Private Endpoint for DBFS Storage Account (OPTIONAL)

You could also optionally provision a private endpoint for the DBFS Storage Account. **There's no separate enablement required for this step.** Please follow the instructions below:

- Find the DBFS storage account for your Azure Databricks workspace through any of these means:
    - Azure Portal
        - In the workspace resource in the portal, go to the Managed Resource Group (note down its name and press the link)



        - In the Managed Resource Group, look for the lone storage account (note down its name starting with "*dbstorage**")



    - Azure CLI
        - Run *az resource show --name <workspace_name> --resource-type "Microsoft.Databricks/workspaces" --resource-group <workspace_parent_resource_group> --subscription <subscription_name>*
        - Note down the values of properties *managedResourceGroupId* (get the name after */resourceGroups/*) and *storageAccountName*

- Create the private endpoint (Portal / CLI) to the DBFS Storage Account
    - Select Resource Type as *Microsoft.Storage/storageAccounts*
    - Select Resource as the DBFS Storage Account (use the above noted names of Managed Resource Group and Storage Account)
    - Select Target sub-resource as *blob*
    - For the VNET and Subnet, we recommend to use a pre-provisioned private endpoint subnet in the workspace VNET itself where it's separate from the workspace subnets. Ideally, this would be the same subnet as you would've used for the Data Plane to Control Plane (Backend) private endpoint created in Step 2.

Note: If this private endpoint is created by a user without CONTRIBUTOR or OWNER permissions on the workspace resource, then somebody with those relevant permissions will need to approve the private endpoint connection.

Once done, all traffic from your clusters to DBFS storage account will use the private endpoint route.