NetBrain® R 12.1

NetBrain Azure Quick Setup Guide

Version 12.1 | Last Updated 2025-03-19

Contents

1. Setup Azure API Access

1.1. Overview

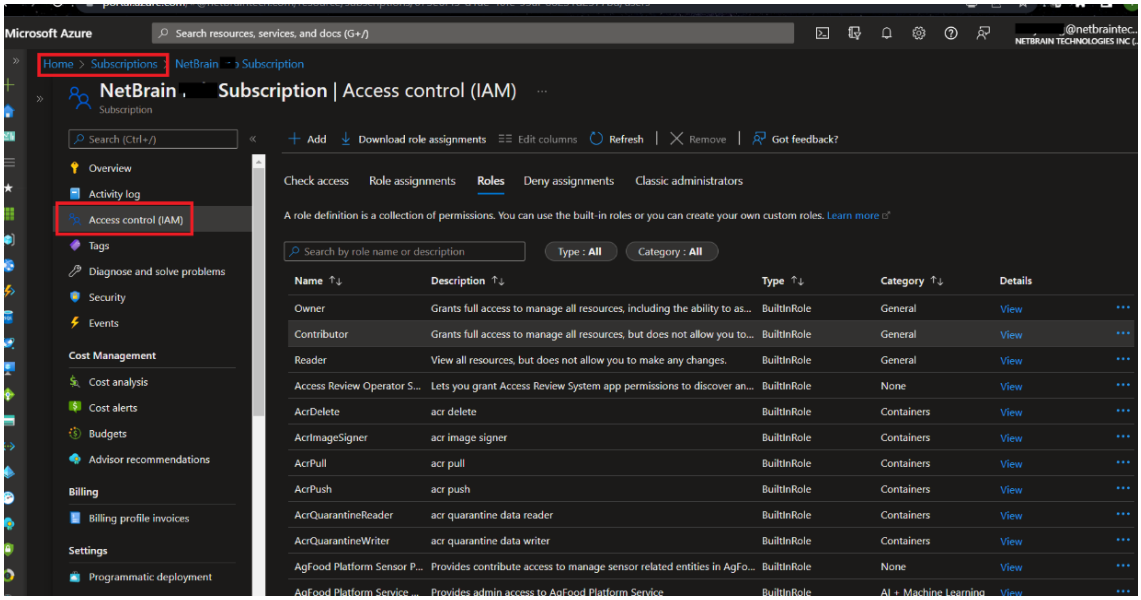
NetBrain accesses Azure through either Managed Identity or Service Principle and uses Rest APIs to retrieve the data from Azure. To enable the NetBrain IE system to retrieve the Azure data, you need to:

1. Create a Custom IAM Role
2. Access APIs (you can select option a or b)
   a) Set Up VM Identity (for Managed Identity)
   b) Register App (for Service Principle)
3. Assign Subscription Role
4. Set up API Server

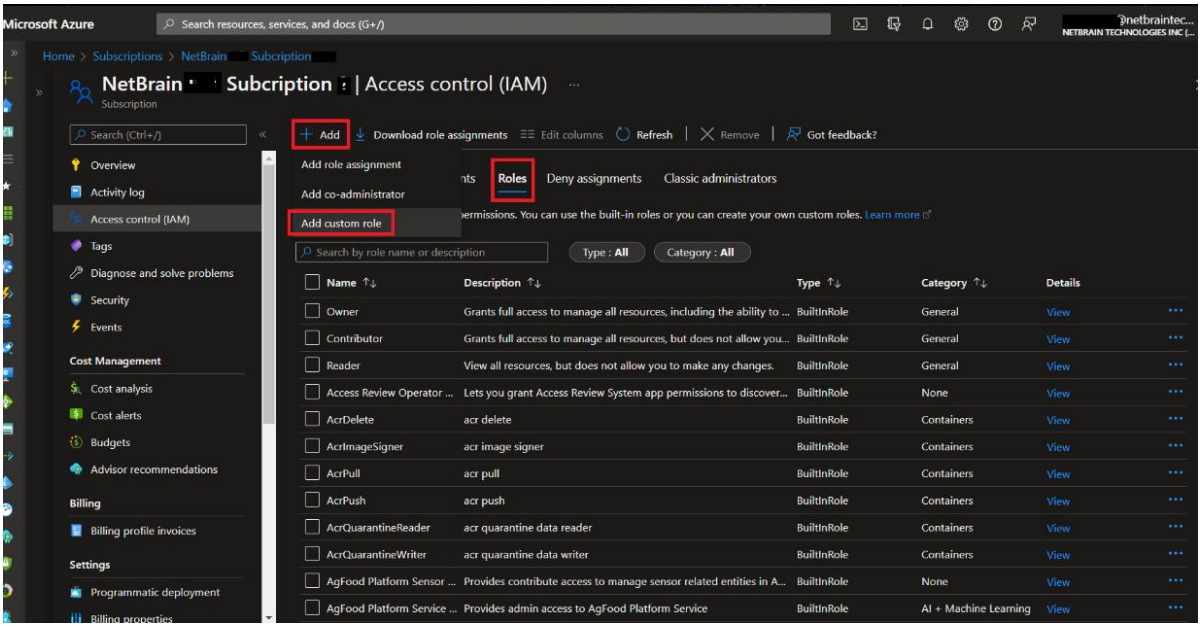
1.2. Create a Custom IAM Role

Azure provides role-based access control (RBAC) to manage access to Azure resources. Follow these steps to create a custom IAM (Identity and Access Management) role for the NetBrain IE system to access Azure APIs:

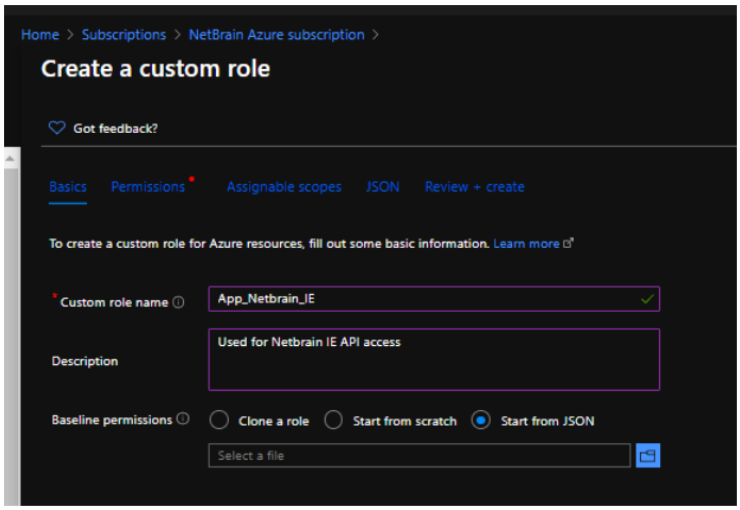
1. Go to **Access control (IAM)** under **Subscriptions** in Azure Portal.



2. Go to **Roles** and click **+Add** and click **New custom role** to create a custom IAM role**.**



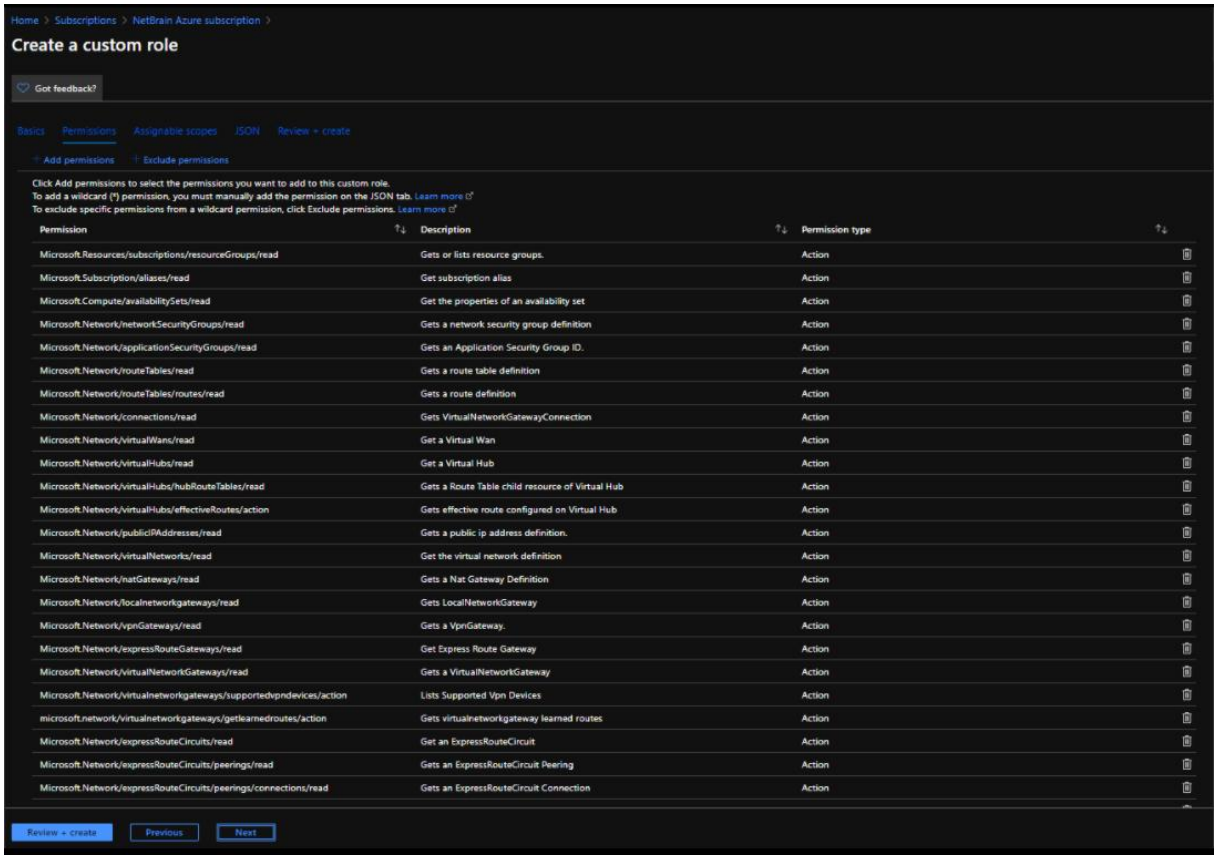
3. Define **Basics** Configuration.



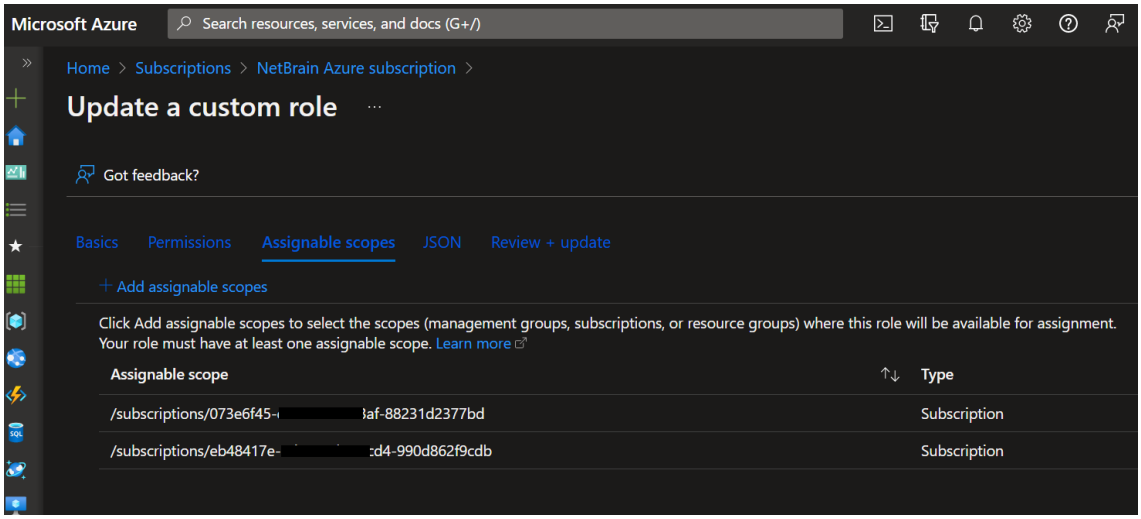
Select **Start from JSON** in the **Baseline permissions** field and import the JSON file below to customize the role with the minimal permissions required for NetBrain IE system discovery and data retrieval.

To get the latest JSON file, refer to NetBrain Online Help: [NetBrain Required Azure IAM Permissions]

4. Review **Permissions**.



5. Select proper subscription or management groups as **Assignable Scopes.**



6. Review Custom Role in JSON Format. Finally, click **Create** button after reviewing the **Review + create** page.

NetBrain supports both **Managed Identity** and **Service Principal** types to access Azure APIs.

- If you select Managed Identity, follow the steps in section 1.3;
- If you select Service Principal, follow the steps in section 1.4.

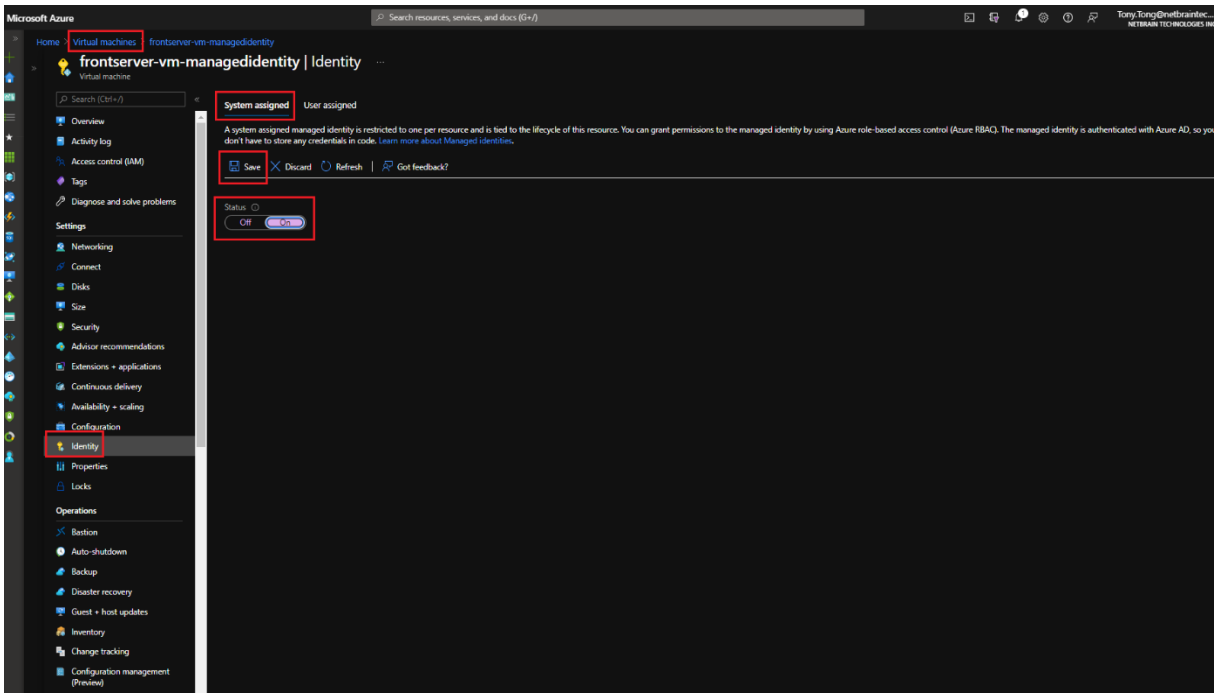
1.3. Access APIs With Managed Identity (Option 1)

1.3.1. Set Up VM Identity

NetBrain supports both Azure-system-assigned and user-assigned Identity for VM, which acts as an application server to access APIs. You only need to choose either one of the following methods below.

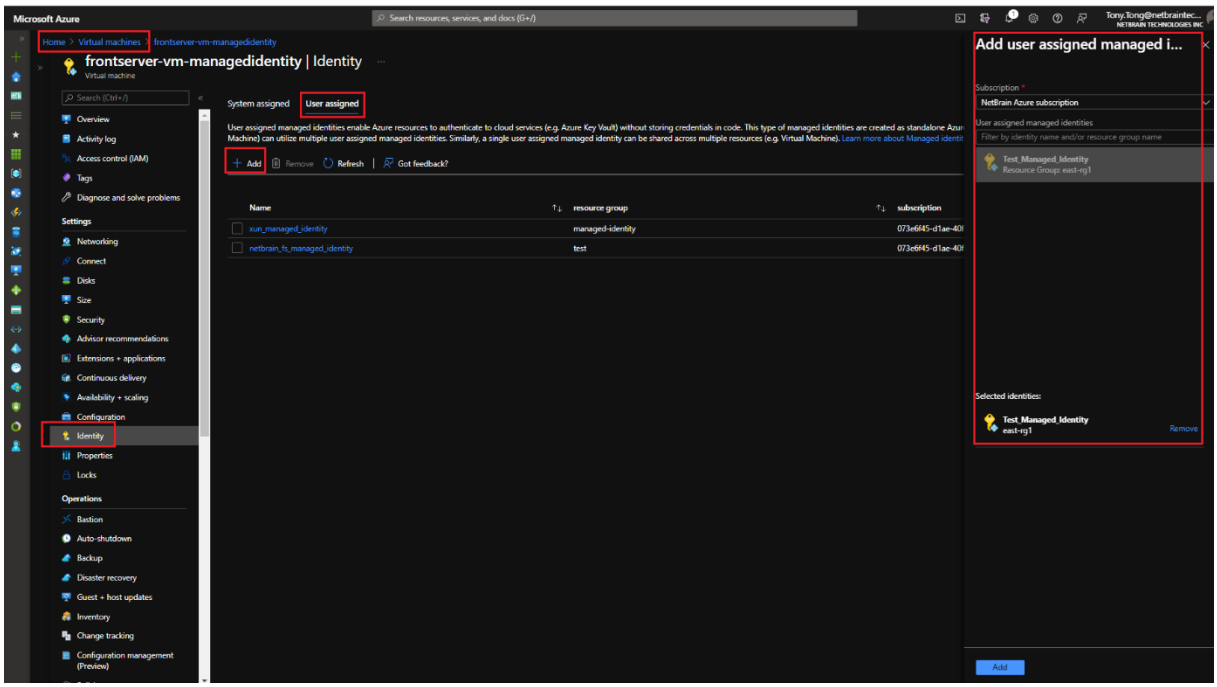
Set Up Azure-system-assigned Identity

1. Go to the **Identity** page of Virtual Machine, which will be used as a NetBrain server from Azure Portal.
2. Select **System assigned menu,** switch to **On** for **Status** and click the **Save** button.



Set Up User-assigned Identity

1. Go to the **Identity** page of Virtual Machine, which will be used as a NetBrain server from Azure Portal.
2. Select the **User assigned** menu and click **+ Add** button. Refer to Azure's official guide to create a managed identity and select it.

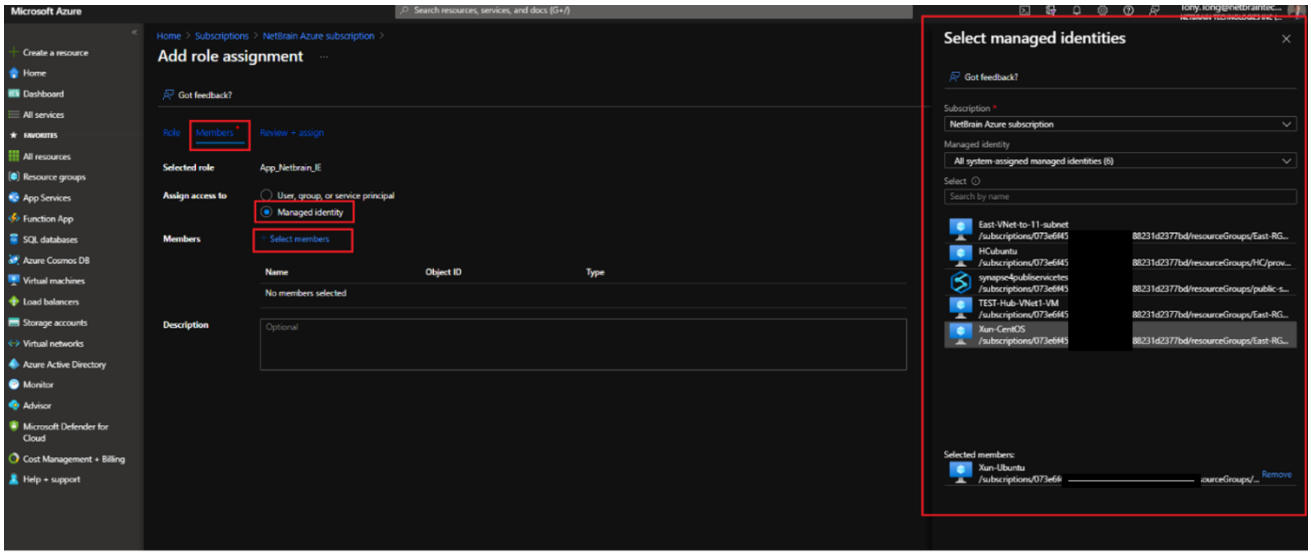


1.3.2. Assign Subscription Role

NetBrain supports both service principal and managed identity as role assignment types. Follow these steps to assign the previously created custom role to managed identity for the NetBrain IE system to access Azure APIs:

1. Go to **Access control (IAM)** within the subscription.

2. Select the previously created custom IAM role, add the new role assignment, and click **+ Select members** to select the previously created VM with managed identity for the NetBrain IE system.

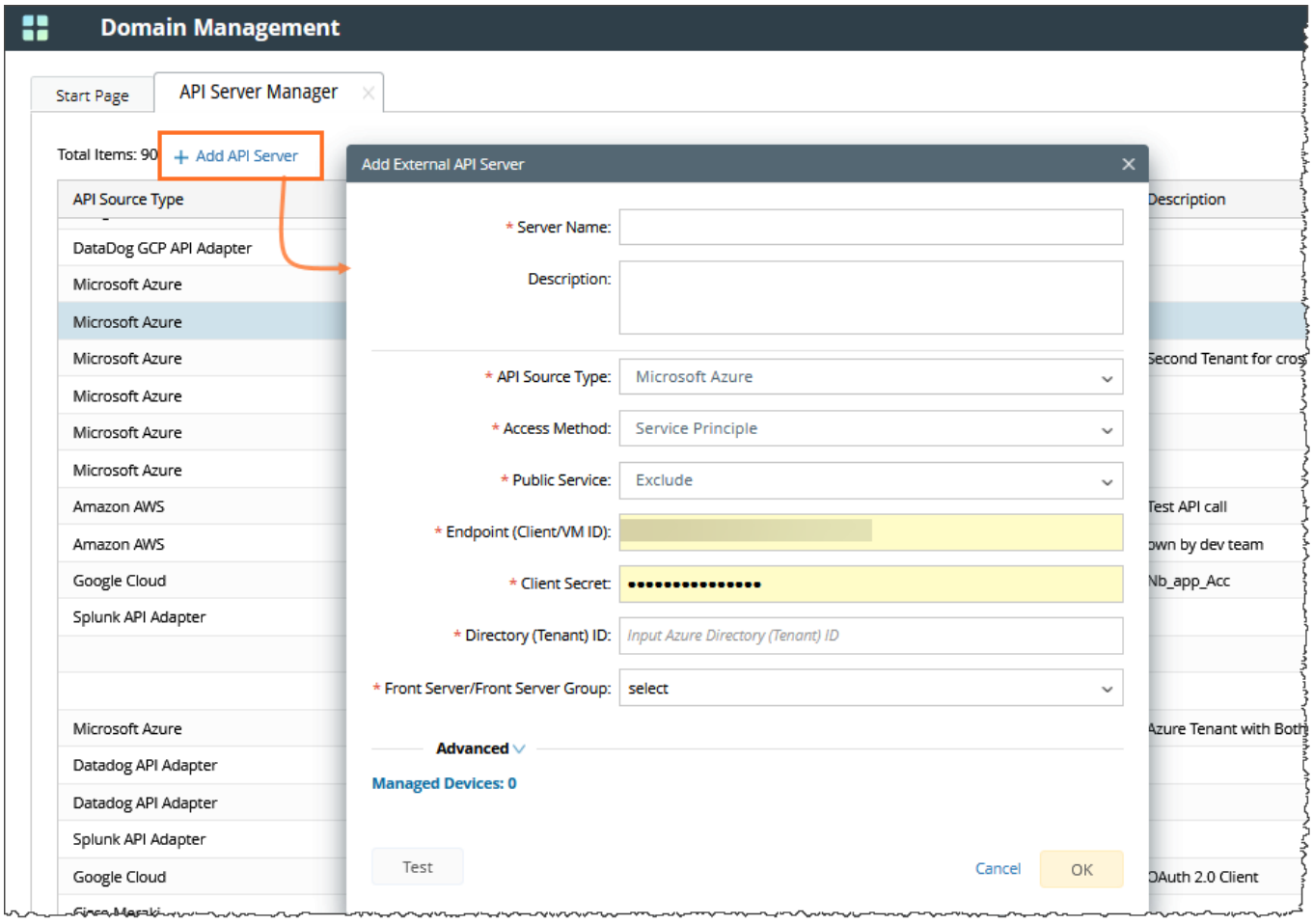


1.3.3. Set Up API Server

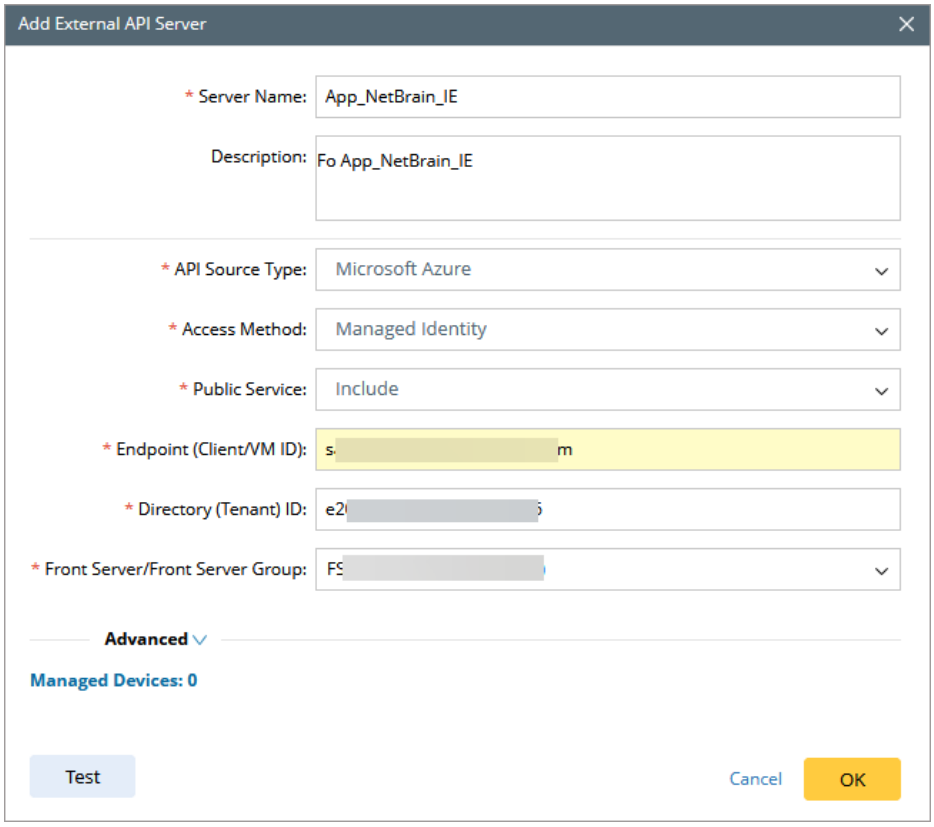
> **Note:** Before setting up your API server, read NetBrain Requirements for API Server Setup first.

Follow these steps to set up a NetBrain API server:

1. Open the **Domain Management** page in the NetBrain IE System. Locate the Quick Access tab, select **API Server Manager**, then click **+ Add API Server** to initiate the addition of a new API server.

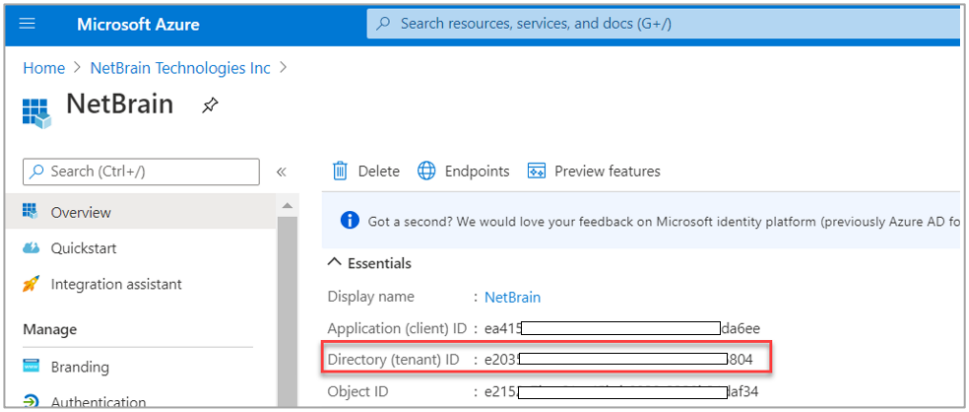


2. Add a new external API Server for Azure access.

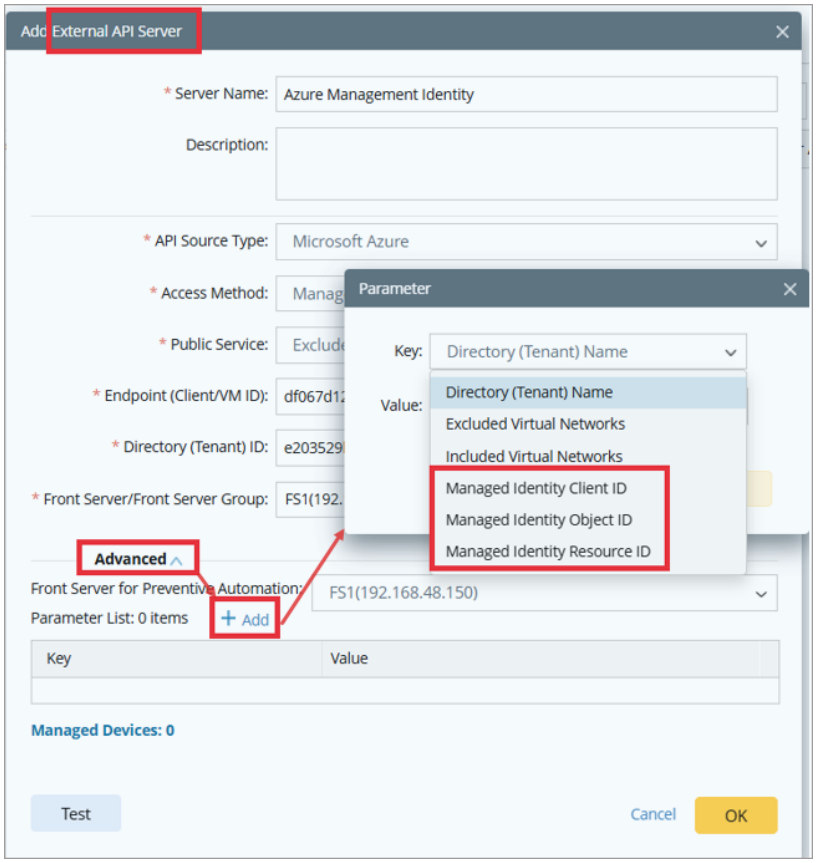


- **Server Name:** a unique name.
- **Description:** an optional field to describe this server.
- **API Source Type**: select **Microsoft Azure**.
- **Access Method**: select **Managed Identity**.
- **Public Service**: Include or Exclude
- **Endpoint (Client/VM ID)**: copy/paste the **VM ID** value from the JSON format data in Azure Virtual Machine (VM) portal.

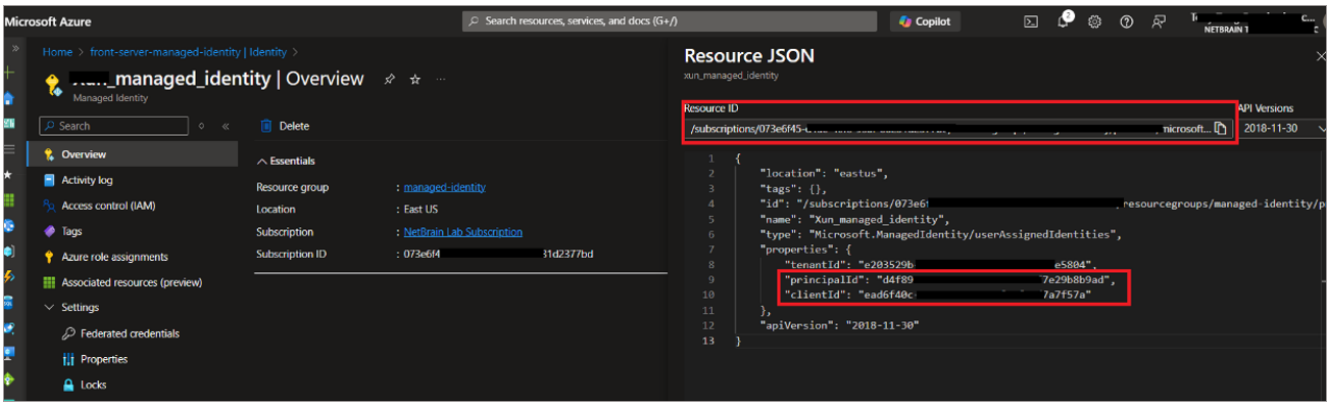
- **Directory (Tenant) ID:** copy/paste the Tenant ID.



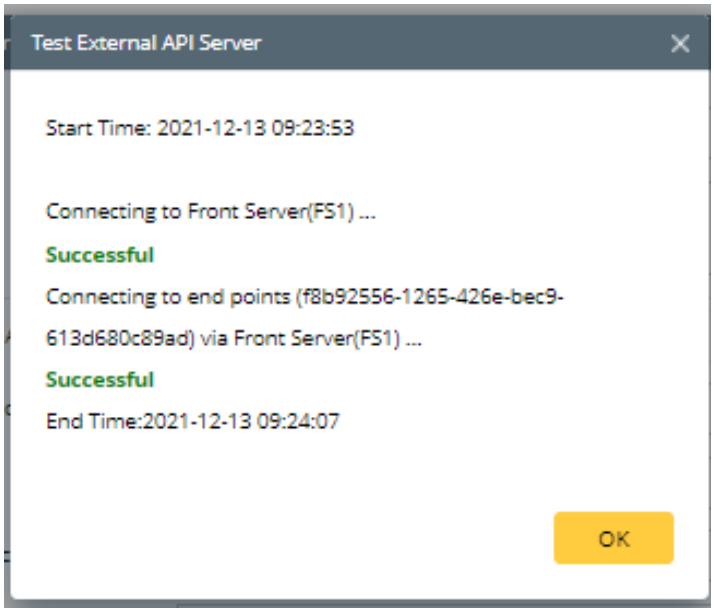
- **Front Server/Front Server Group**: select one front server VM with managed identity to access Azure.
- **Management Identity Client ID/Object ID/Resource ID**: In the **Advanced** settings, you can specify which VM management identity will be used for NetBrain IE discovery.
    - If the VM is configured with only a ***system-assigned identity*** or if the VM has only ***one user-assigned identity*** configured, no additional configuration is required.
    - If both identity types are configured and you prefer to use the ***user-assigned identity*** (note: Azure defaults to the ***system-assigned identity***). To override the default ***system-assigned identity*** and use a **user-assigned identity**, you must explicitly define the ***management identity*** using one of the following parameters: **Client ID**, **Object ID**, and **Resource ID.**



- Find the management Client ID, Object ID(PrincipalId) or Resource ID from Json view of Azure Portal.



3. Click **Test** to verify that this API server works.



Once the API server is successfully verified and saved, you can proceed to Discover Azure resources.

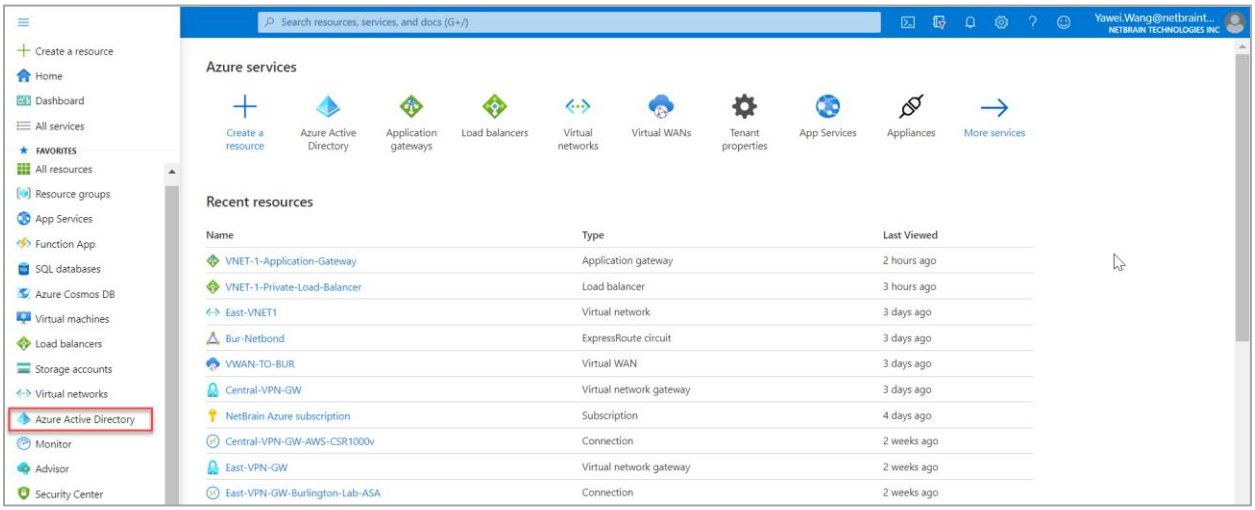
1.4. Access APIs With Service Principal (Option 2)

1.4.1. Register App

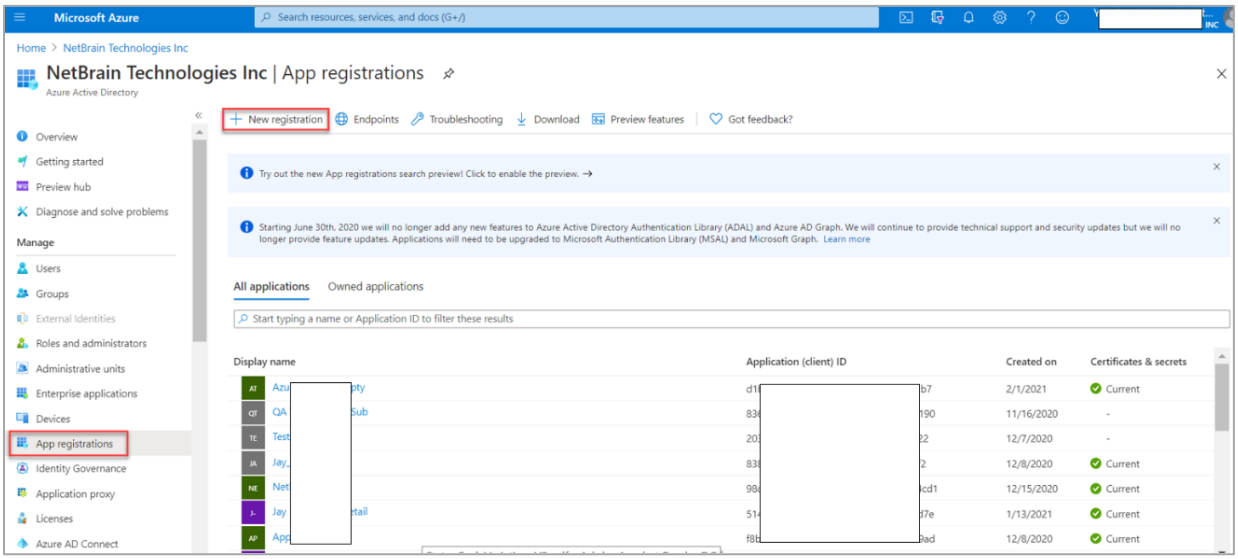
The Microsoft identity platform performs IAM only for registered applications. Therefore, an App must be registered in the Azure portal to establish a trusting relationship between NetBrain IE and Azure.

Follow these steps to register an App for the NetBrain IE system:

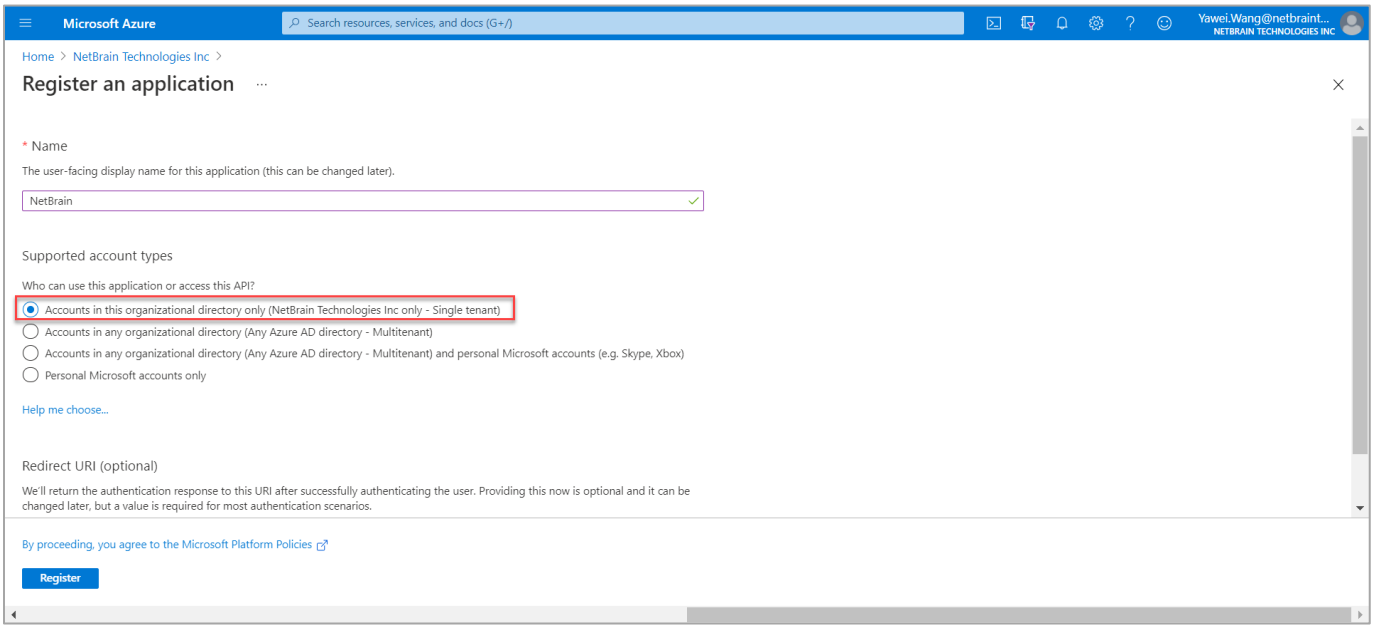
1. Go to **Azure Active Directory** in Azure Portal.



2. Go to **App registrations** and click **New registration**.

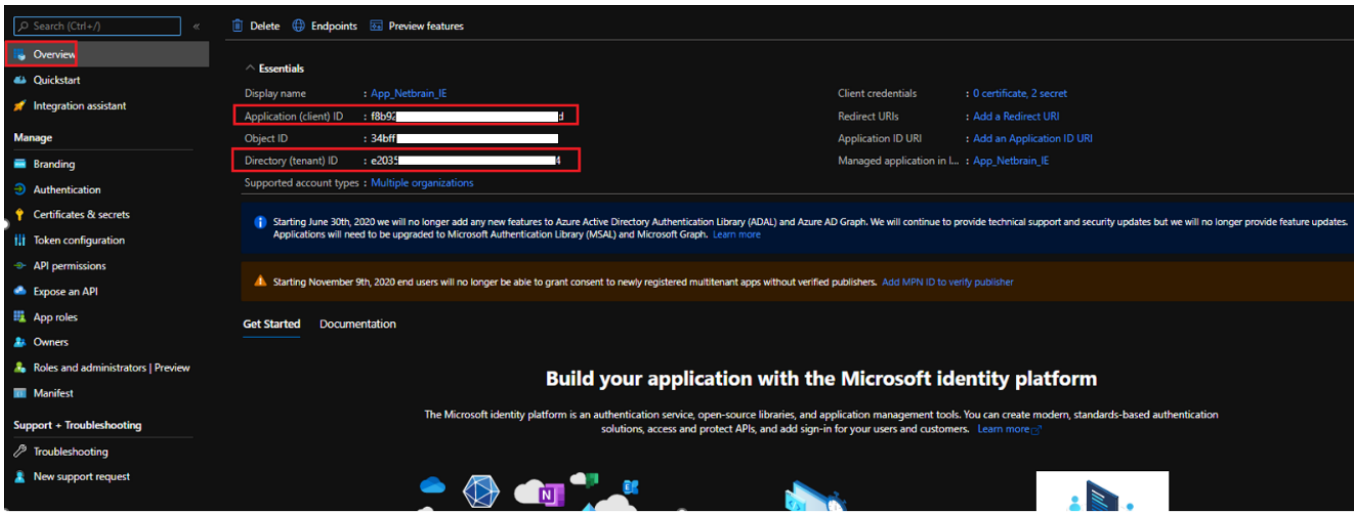


3. Define an App **name**, select the account type Accounts in this organizational directory only (Single tenant), and then click **register**.

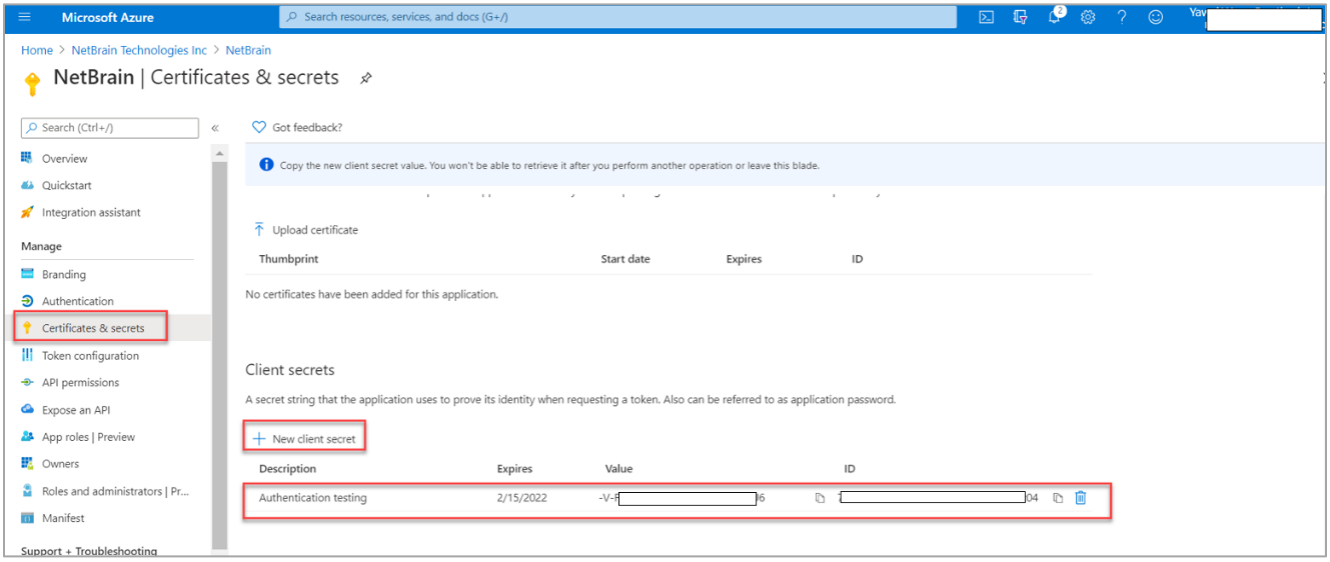


> **Note:** NetBrain only fully supports the "Single Tenant" account type.

4. Go to the **Overview** page of the newly registered App. The **Application (Client) ID** and **Directory (Tenant) ID** information will be used to set up the NetBrain external API server later**.**



5. Go to **Certificate & secrets** within the created App to add a new **client secret**. The value of the newly created client secrets will be used to set up the NetBrain external API server later.

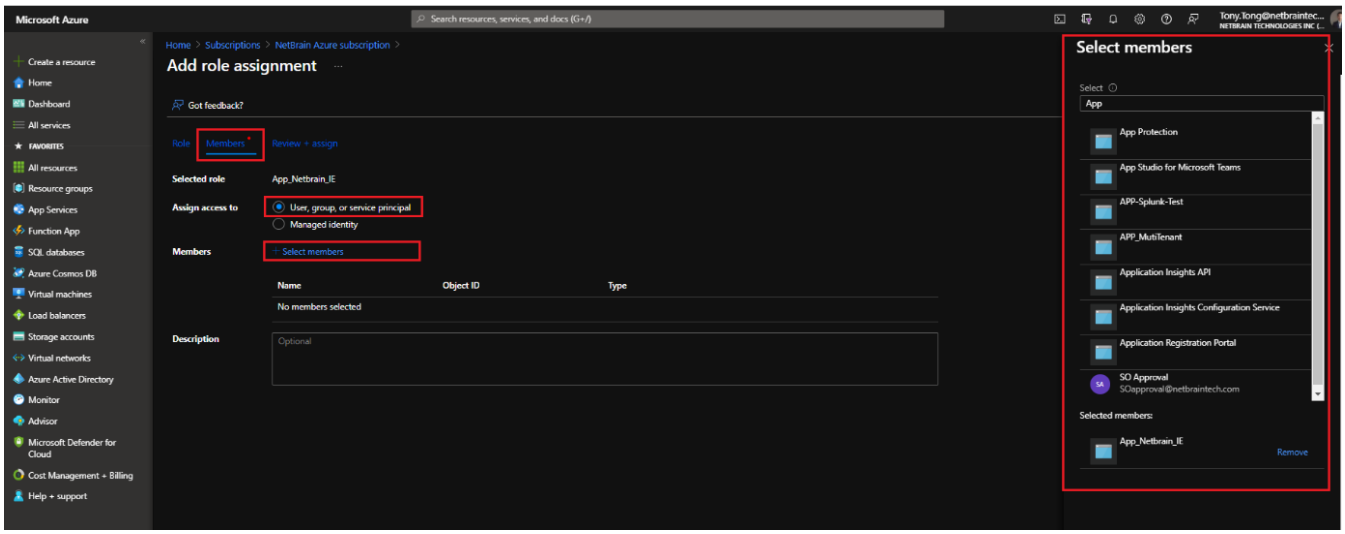


1.4.2. Assign Subscription Role

NetBrain supports both service principal and Managed identity as role assignment targets. Follow these steps to assign the previously created custom role to the service principal for the NetBrain IE system to access Azure APIs:

1. Go to **Access control (IAM)** within the subscription.

2. Select the previously created custom IAM role, add a new role assignment, and click **+ Select members** to select the previously registered Application for the NetBrain IE system.

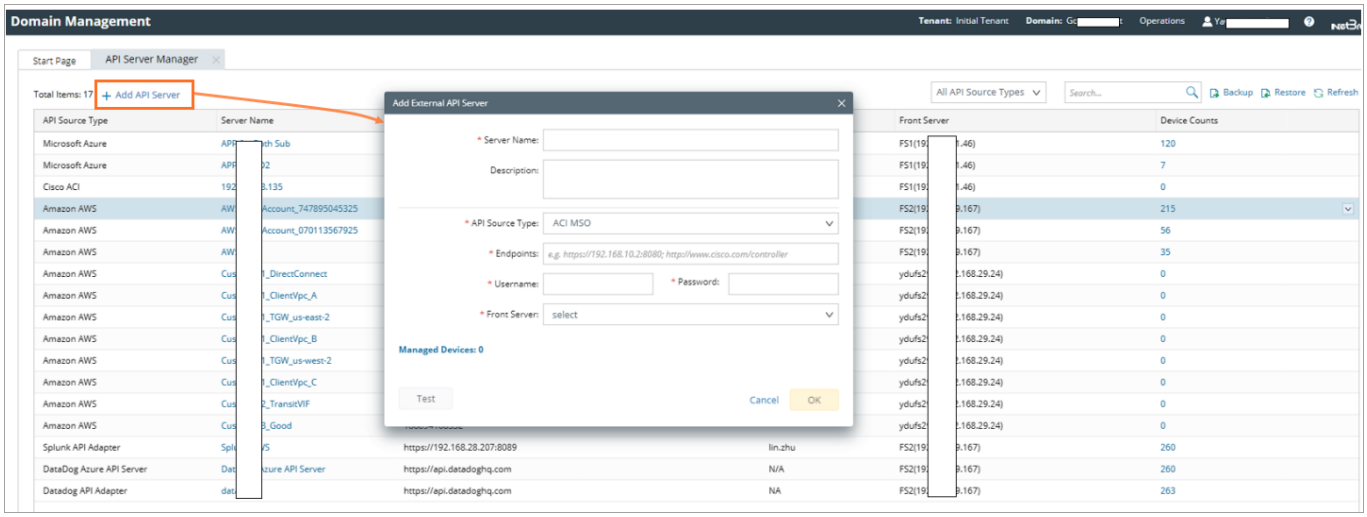


1.4.3. Set Up API Server

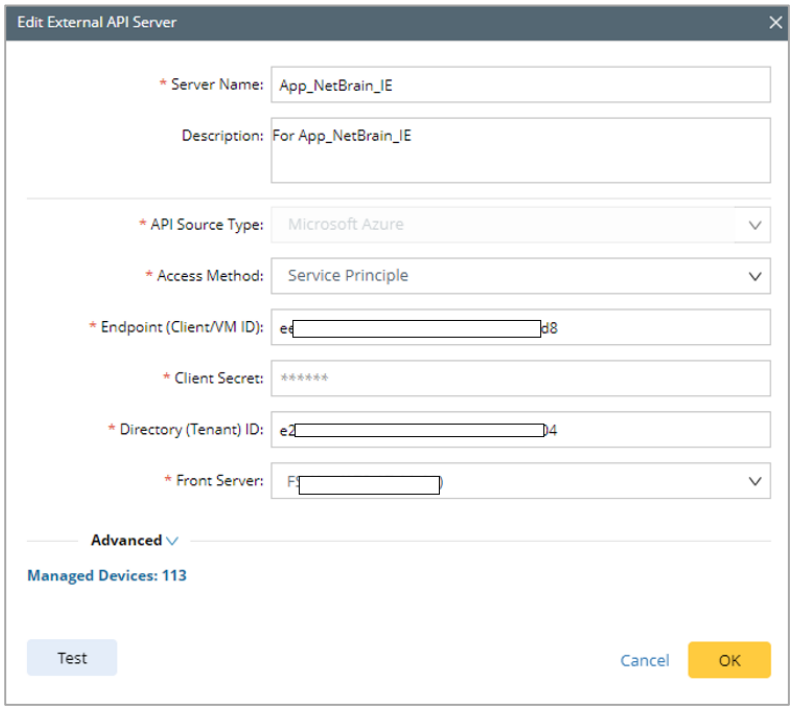
> **Note:** Before setting up your API server, read NetBrain Requirements for API Server Setup first.

Once you have completed the steps above, follow these steps to set up a NetBrain API Server:

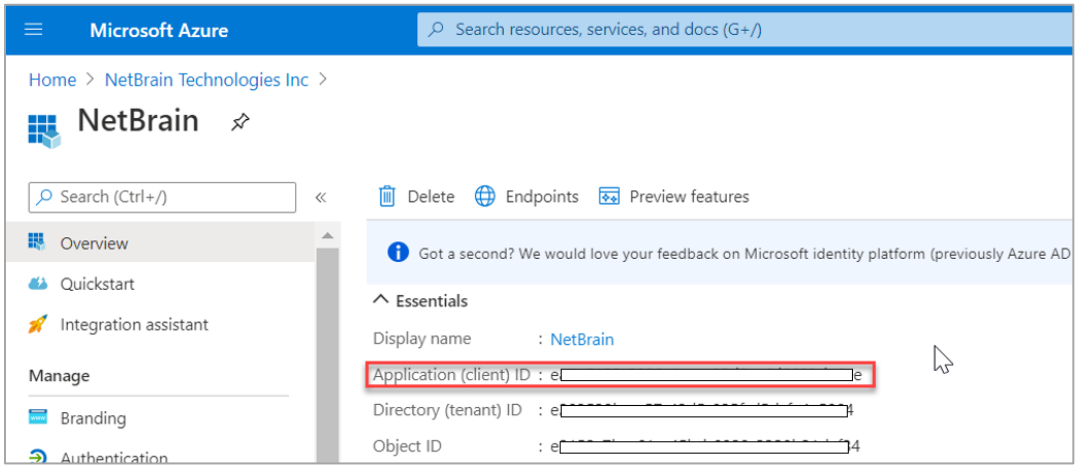
1. Open the **Domain Management** page of the NetBrain IE system, select **Operations > Discover Settings > API Server Manager** from the quick access tab, and click **Add API server**.



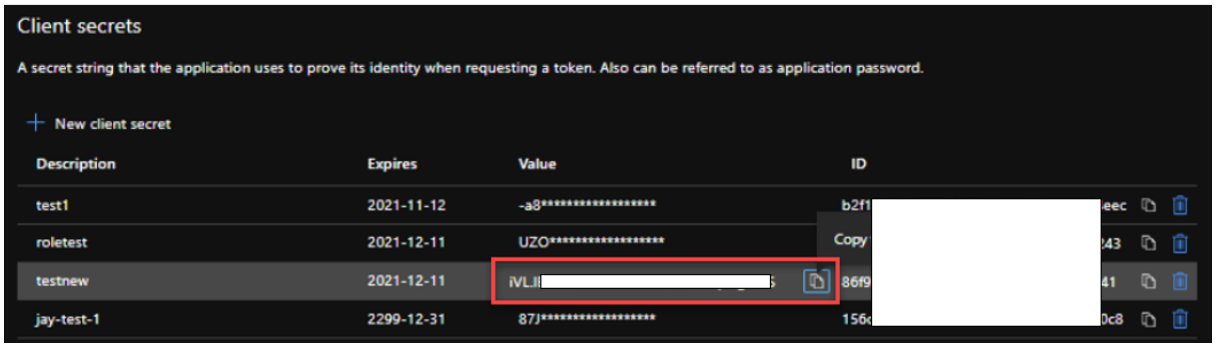
2. Add a new external API Server for Azure access.



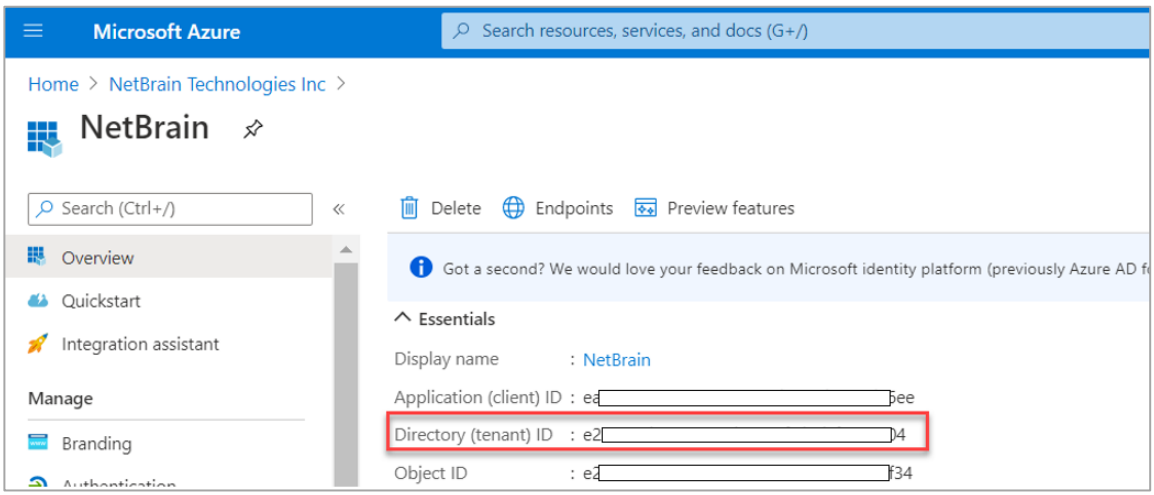
- **Server Name:** a unique name.
- **Description:** an optional field to describe this server.
- **API Source Type**: select **Microsoft Azure**.
- **Access Method**: select **Service Principle**.
- **Endpoint (Client/VM ID)**: copy/paste the ID from your registered App.



- **Client Secret**: copy/paste the value from created client secret within your registered App.

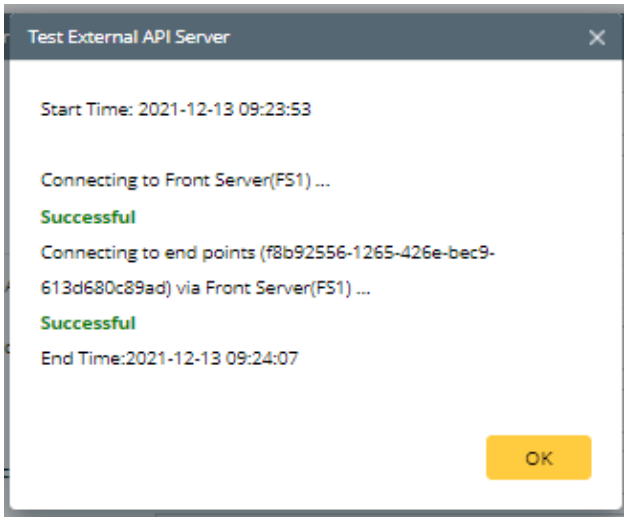


- **Directory (Tenant) ID**: copy/paste the ID from your registered App.



- **Front Server**: select one front server which can access Azure.

3. Click **Test** to verify that the API server works.

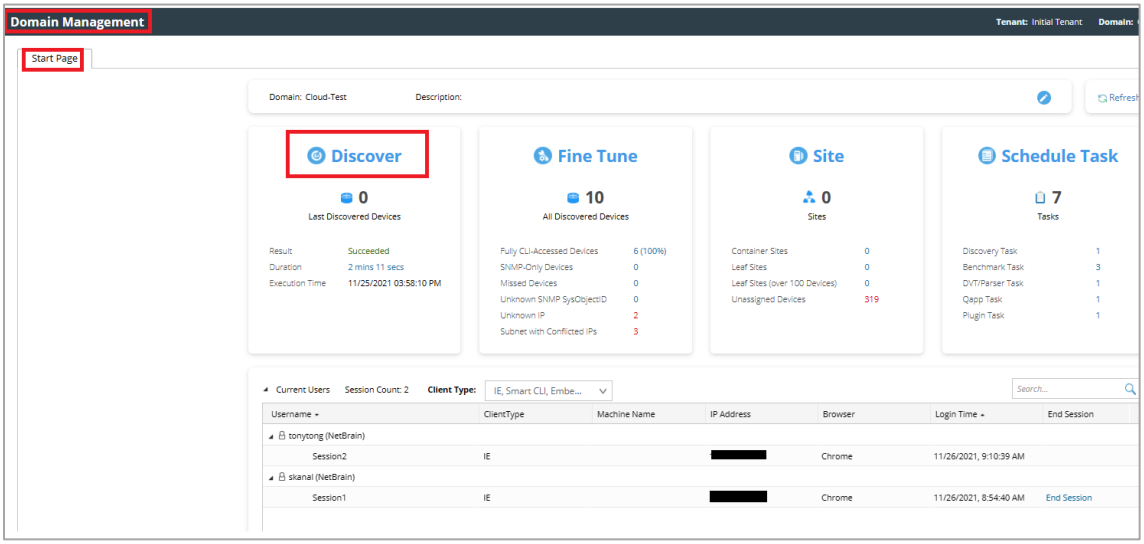


Once the API server is successfully verified and saved, you can proceed to Discover Azure resources.

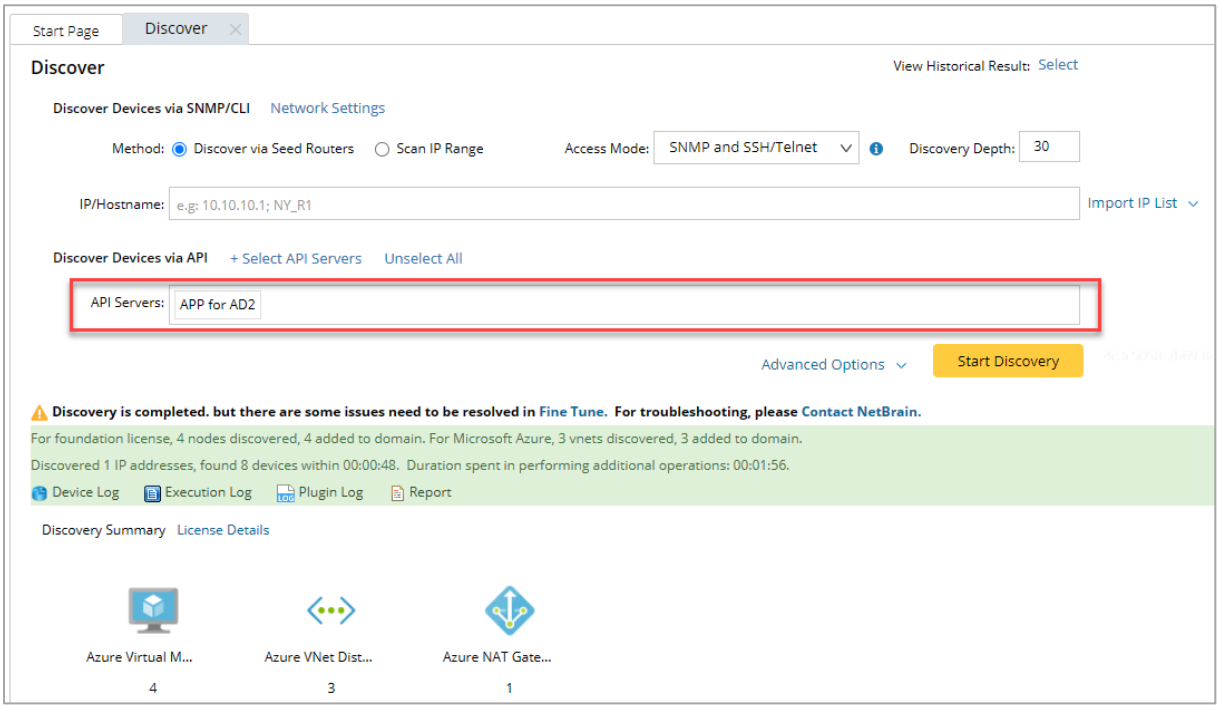
2. Discover Azure Resources

Follow these steps to discover Azure networking resources via APIs:

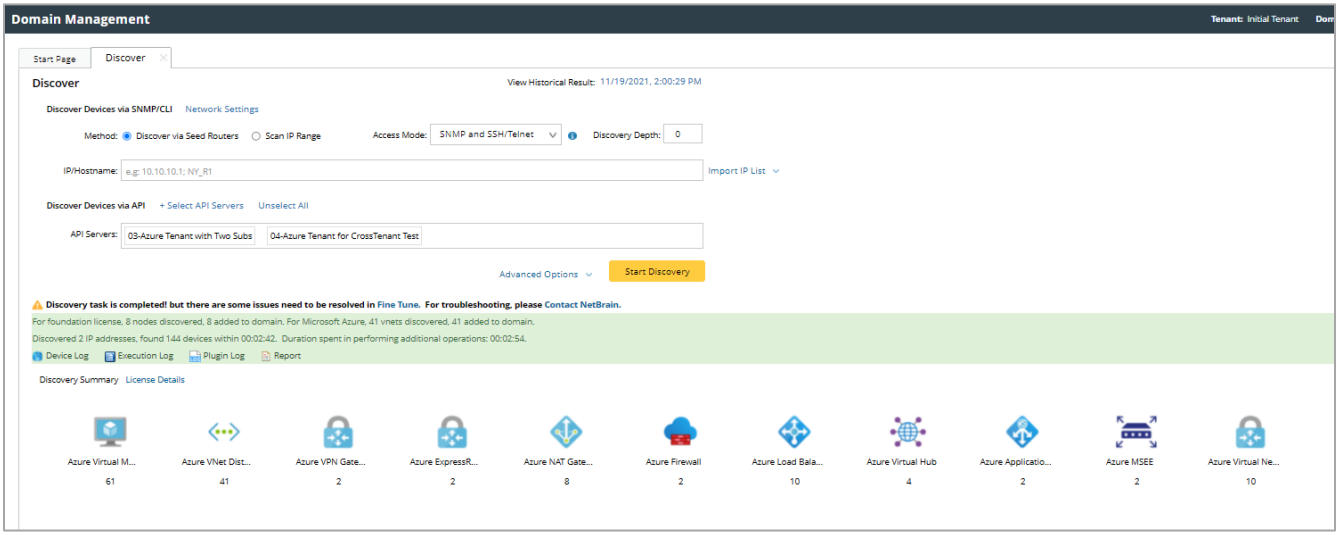
1. Log in to your NetBrain IE system.
2. Click the **Discover** link from the **Domain Management** page.



3. Click **+ Select API Servers** and select the Azure API Server you just created.
4. Click **Start Discovery** to start the Azure discovery. Run the discovery task manually or schedule a discovery task to discover Azure resources.



5. Wait till the discovery finishes. The number of discovered objects will be displayed.

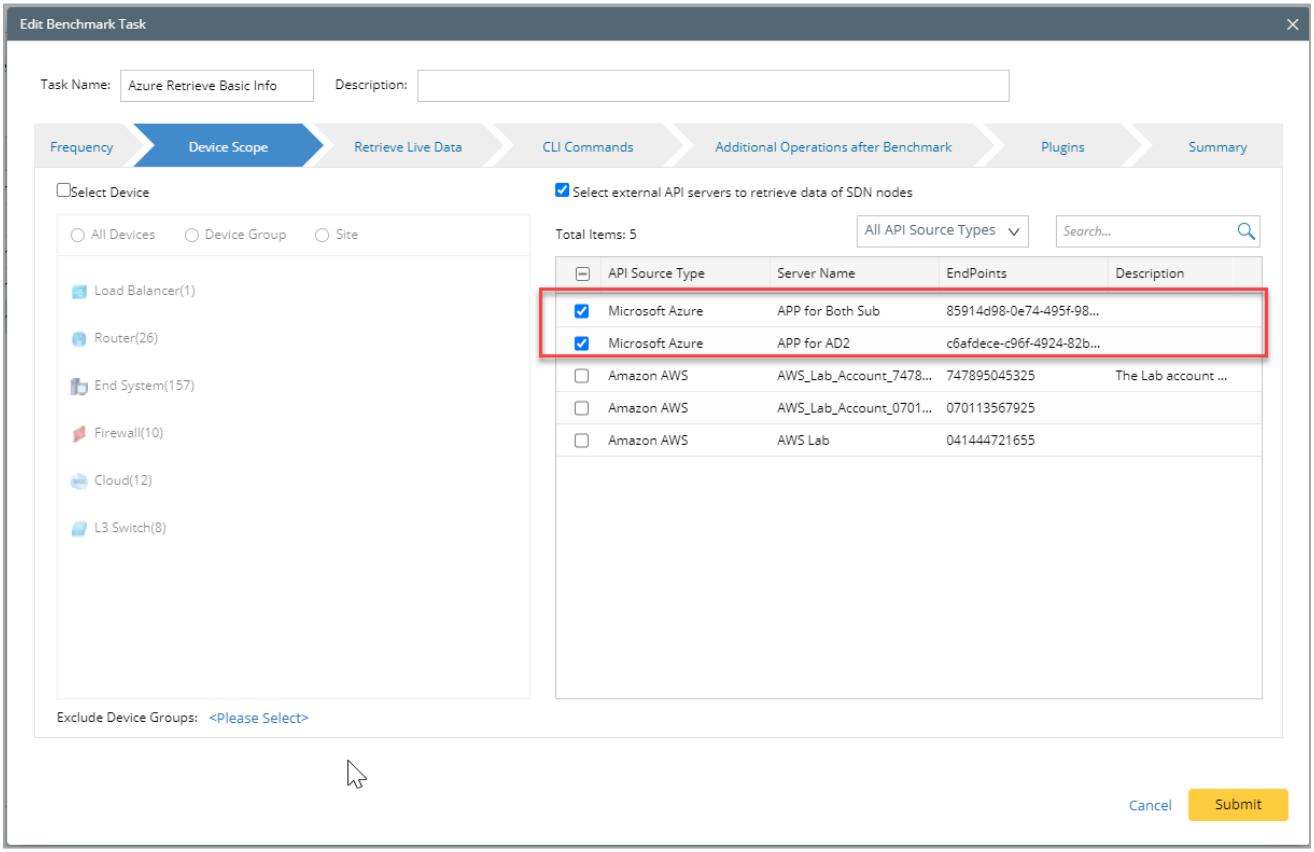

Domain Management
Tenant: Initial Tenant
Start Page
Discover
Discover
View Historical Result: 11/19/2021, 2:00:29 PM
Discover Devices via SNMP/CLI
Network Settings
Method: Discover via Seed Routers
Scan IP Range
Access Mode: SNMP and SSH/Telnet
Discovery Depth: 0
IP/Hostname: e.g. 10.10.10.1; NY_R1
Import IP List
Discover Devices via API
+ Select API Servers
Unselect All
API Servers: 03-Azure Tenant with Two Subs
04-Azure Tenant for CrossTenant Test
Advanced Options
Start Discovery
Discovery task is completed! but there are some issues need to be resolved in Fine Tune. For troubleshooting, please Contact NetBrain.
For foundation license, 8 nodes discovered, 8 added to domain. For Microsoft Azure, 41 vnets discovered, 41 added to domain.
Discovered 2 IP addresses, found 144 devices within 00:02:42. Duration spent in performing additional operations: 00:02:54.
Device Log
Execution Log
Plugin Log
Report
Discovery Summary
License Details
Azure Virtual M... 61
Azure VNet Dist... 41
Azure VPN Gate... 2
Azure ExpressR... 2
Azure NAT Gate... 8
Azure Firewall 2
Azure Load Bala... 10
Azure Virtual Hub 4
Azure Applicatio... 2
Azure MSEE 2
Azure Virtual Ne... 10

3. Auto-Update and Enrich Azure Data

The discovery process only retrieves the basic data of your Azure network and builds L3 topology. After the discovery, you need to set up a NetBrain benchmark task to retrieve all data, including visual spaces and data views.

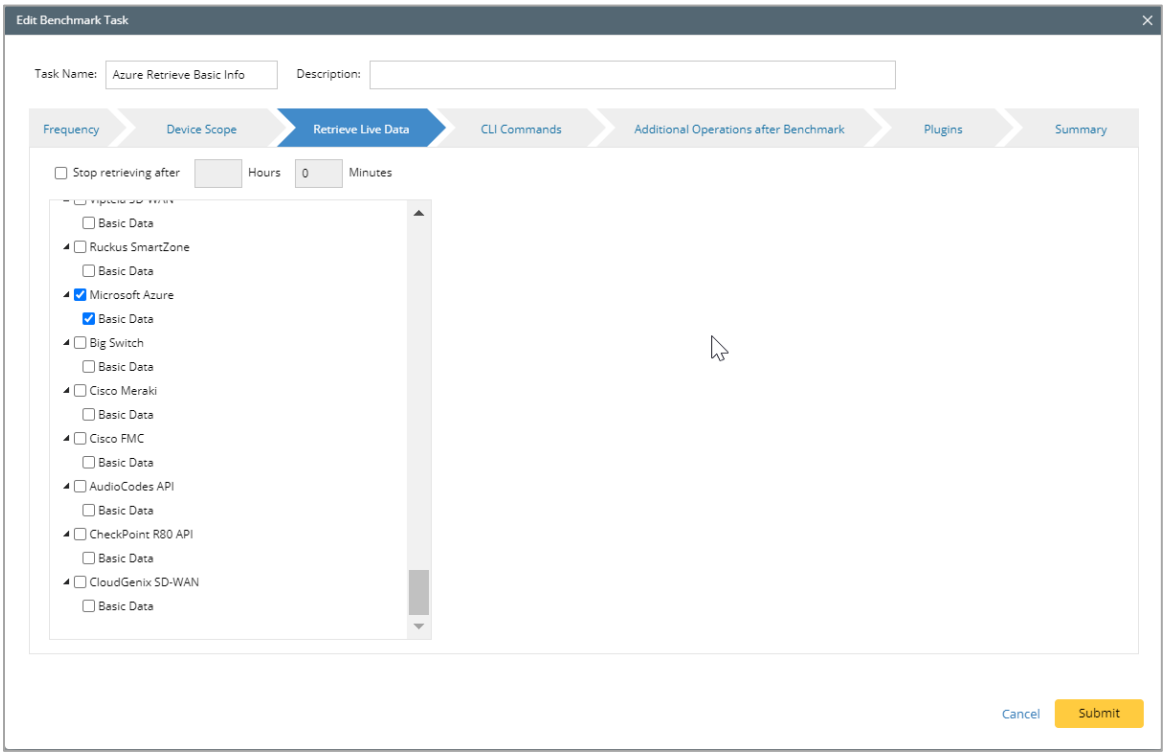
To create a benchmark for Azure resources, complete the following steps.

1. On the Start Page, click the **Schedule Task** link.
2. On the **Schedule Discovery/Benchmark** tab, click **Add Benchmark Task**.
3. On the **Frequency** tab, define the task frequency.
4. On the **Device Scope** tab, check the **Select external API servers to retrieve data** check box and select the API server for Azure.



> **Note:** It's highly recommended to re-use the "Basic System Benchmark" with a full benchmark task to ensure all Azure-connected physical or virtual devices are selected within the device scope.

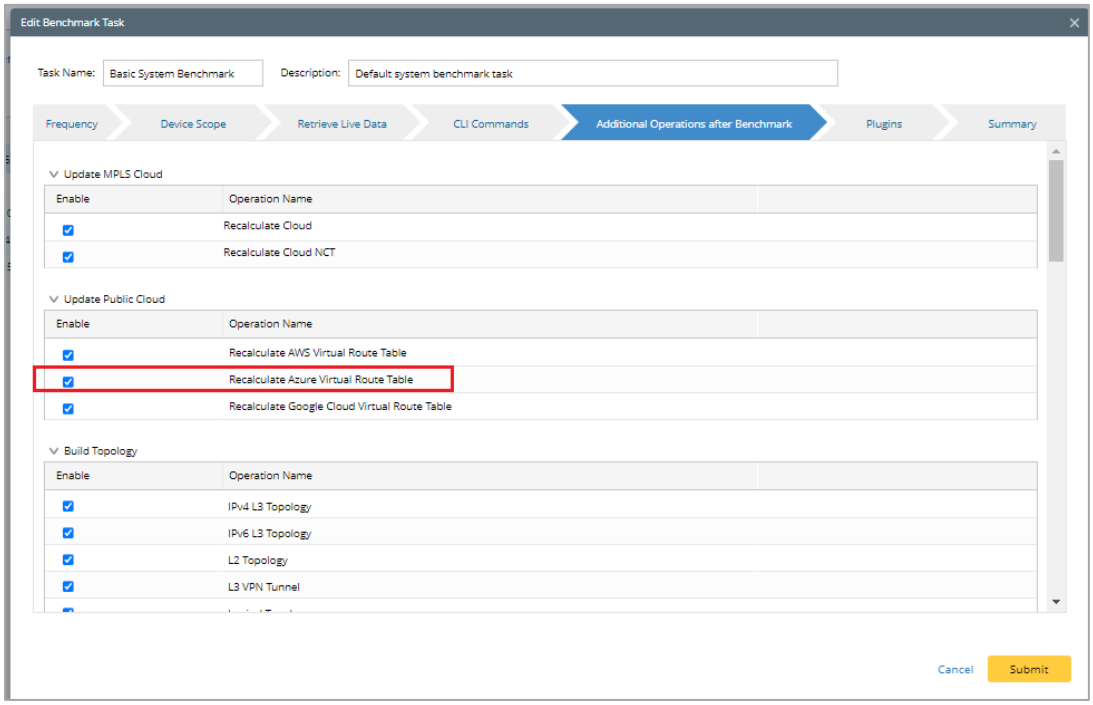
5. On the **Retrieve Live Data** tab, select the **Microsoft Azure Basic Data** check box, keep the default selected NCT tables as they are, and select **BGP Advertised Route Table**.



6. On the **Additional Operation After Benchmark** tab, select the check boxes for:

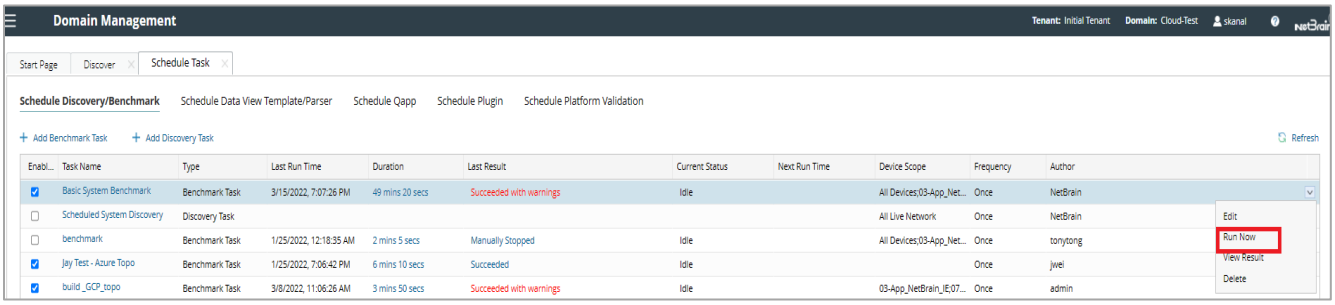
   - Update MPLS Cloud
   - Update Public Cloud (Recalculate Azure Virtual Route Table)

- Update Build Topology

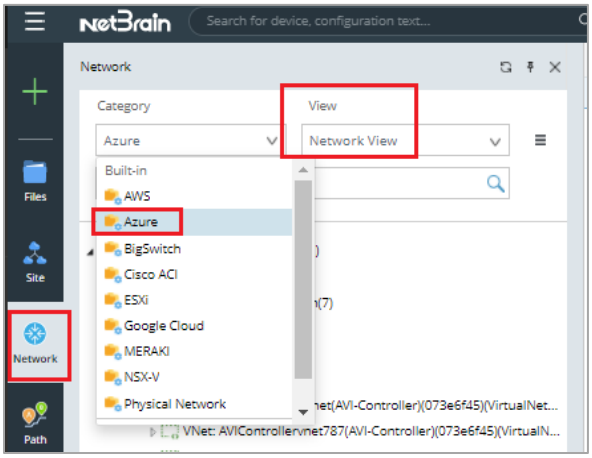


7. Click **Submit**.

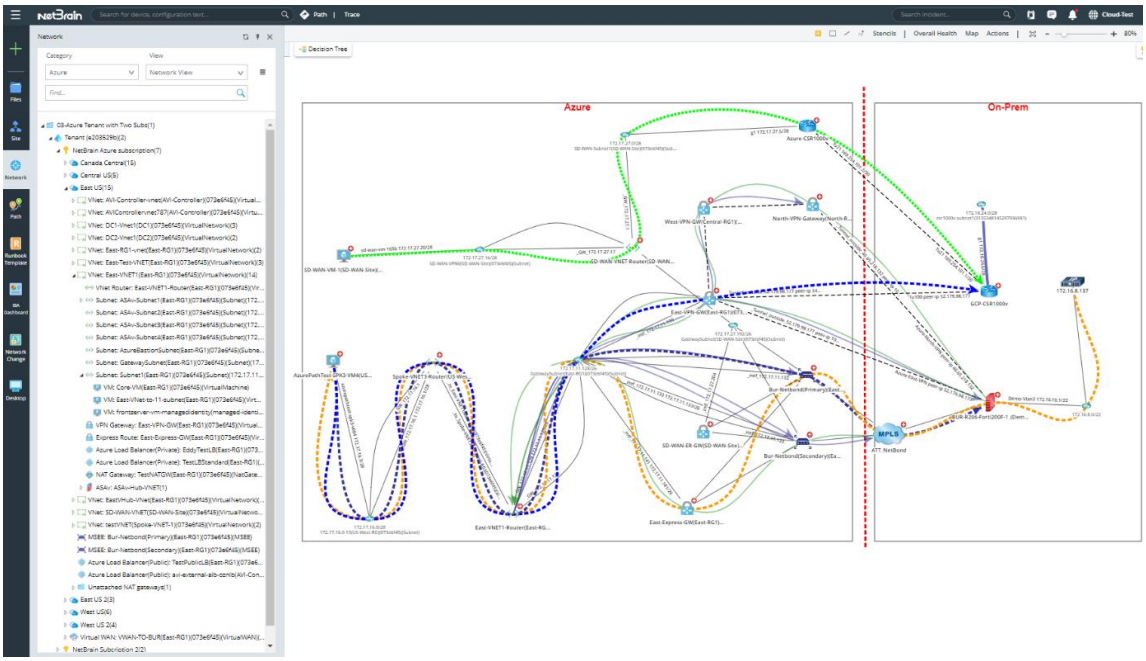
8. Trigger the benchmark task by clicking **Run Now**.



9. Open the network tree and select **Azure** in the **Category** field to view the Azure resource. Select **Network View** in the View field.



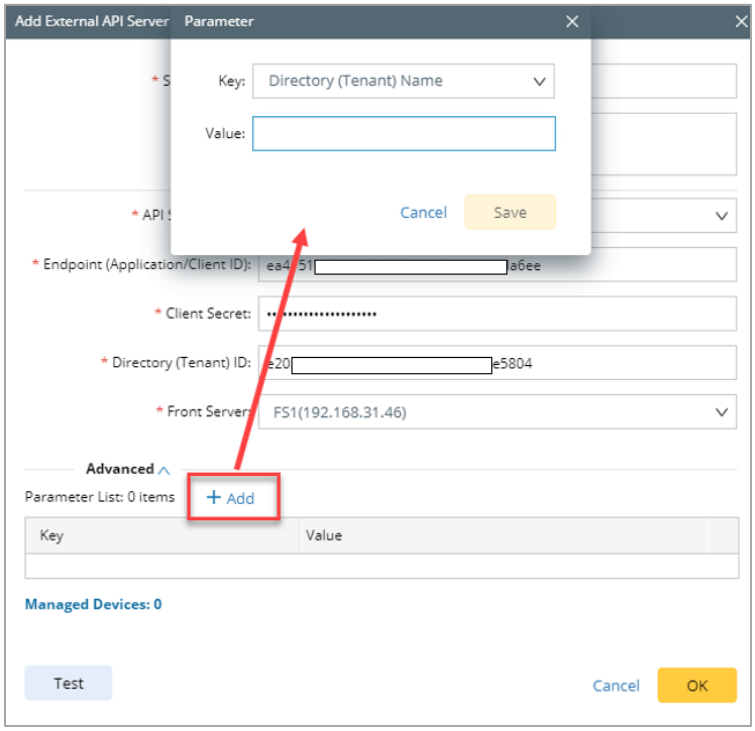
All Azure resources discovered are displayed on the network tree. In addition, you can click a resource to open its context map.



4. Appendix

4.1. NetBrain Requirements for API Server Setup

- The minimum resource unit for an API Server scope is a subscription that includes all the resources under it. Therefore, NetBrain does not recommend separating resources under one subscription to a different API Server.
- The API Server is associated with an AD Tenant. Use IAM to control the subscription level of resources API discovery. If you have multiple AD Tenants, set up at least one API Server for each AD Tenant.
- NetBrain does not retrieve tenant details (including tenant name) using Azure Management APIs but instead creates a random tenant name. However, as below, you can manually define your Tenant Name in the API server manager:



- To collect data from Azure successfully, NetBrain Front Server must have access to ***.core.windows.net**, ***.azure.com**, ***.microsoft.com**, and ***.microsoftonline.com**.

4.2. NetBrain Required Azure IAM Permissions

```
{
  "properties": {
    "roleName": "Role_App_Netbrain_IE",
    "description": "Used for Netbrain IE API access. /action permissions are used due
to Azure design for some processes like downloading some tables which are generated at
run-time.",
    "assignableScopes": [
      "/subscriptions/modifytoyoursubscription1",
      "/subscriptions/modifytoyoursubscription2"
    ]
  },
  "permissions": [
    {
      "actions": [
        "Microsoft.Resources/tenants/*/read",
        "Microsoft.Resources/subscriptions/*/read",
        "Microsoft.Subscription/*/read",
        "Microsoft.Network/*/read",
        "Microsoft.Compute/*/read",
        "Microsoft.Insights/Metrics/*/Read",
        "Microsoft.DocumentDB/databaseAccounts/*/Read",
        "Microsoft.DBforMariaDB/servers/*/Read",
        "Microsoft.DBforMySQL/servers/*/Read",
        "Microsoft.DBforPostgreSQL/servers/*/Read",
        "Microsoft.Sql/servers/*/Read",
        "Microsoft.Storage/storageAccounts/*/read",
        "Microsoft.CostManagement/*/read",
        "PaloAltoNetworks.Cloudngfw/*/read",
        "Microsoft.ResourceHealth/*/read"

        "Microsoft.Network/applicationGateways/backendhealth/action",
        "Microsoft.Network/applicationGateways/getBackendHealthOnDemand/action",
        "Microsoft.Network/applicationGateways/effectiveNetworkSecurityGroups/action",
        "Microsoft.Network/applicationGateways/effectiveRouteTable/action",
        "Microsoft.Network/networkInterfaces/effectiveRouteTable/action",
        "Microsoft.Network/networkInterfaces/effectiveNetworkSecurityGroups/action",
        "Microsoft.Network/virtualHubs/effectiveRoutes/action",
        "Microsoft.Network/virtualNetworkGateways/getadvertisedroutes/action",
        "Microsoft.Network/virtualNetworkGateways/getbgppeerstatus/action",
        "Microsoft.Network/virtualNetworkGateways/getlearnedroutes/action",
        "Microsoft.Network/virtualNetworkGateways/supportedvpndevices/action",
        "Microsoft.Network/virtualwans/vpnconfiguration/action",
        "Microsoft.Network/vpnGateways/getLearnedRoutes/action",
        "Microsoft.Network/vpnGateways/getAdvertisedRoutes/action",
        "Microsoft.Network/virtualNetworkGateways/read",
        "Microsoft.Network/virtualHubs/bgpConnections/advertisedRoutes/action",
        "Microsoft.Network/virtualHubs/bgpConnections/learnedRoutes/action"
      ],
      "notActions": [],
      "dataActions": [],
      "notDataActions": []
    }
  ]
}
```